SAMPLE-EFFICIENT REINFORCEMENT LEARNING

BY

HARSH GUPTA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Professor Rayadurgam Srikant, Chair
Professor Bruce Hajek
Associate Professor Maxim Raginsky
Assistant Professor Niao He

ABSTRACT

Reinforcement learning has been instrumental in the recent advances made by artificial intelligence agents in various domains. Most of these advances have been abetted by the availability of huge amounts of training data. But, in several practical applications such as those arising in wireless networks, robotics, self-driving cars etc., it is expensive and sometimes completely infeasible to collect very large amounts of data. In this work, we study four different such model-free reinforcement learning problems. The first problem we consider is the structured multi-armed bandits problem, motivated by an application in wireless networks. The second problem we consider is the bandits with two-level feedback problem, motivated by an application in panoramic video streaming. The third problem we consider is the analysis of two-time scale reinforcement learning algorithms and the final problem we consider is the analysis of the Double Q-learning algorithm. In each of these problems, our general goal is to theoretically understand the mechanics of the different moving parts in the problem and on the basis of the insights obtained from the theory, design principled practical algorithms/heuristics that are sample-efficient.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to extend my heartfelt gratitude to my family - parents, brother, sister-in-law, Kittu and Cheeku. I have always received generous support and love from my family, especially from my parents, even when they have been facing their own challenges. With the numerous struggles that came with pursuing graduate studies in a new country, it was always comforting for me to have both the implicit and explicit backing of my family.

Over the course of my PhD, I was fortunate enough to make several new friends and also strengthen my bond with many existing friends. These friendships made the academic and personal challenges not only bearable, but also fun at times. Special shout-outs to: friends in CSL 104 - Joseph Lubars, Sai Kiran Burle, Seo Taek Kong, Siddhartha Satpathi, Suryanarayana Sankagiri and Zhenzhe Zheng; other friends in CSL - Amish Goel, Ashok Vardhan Makkuva, Konik Kothari, Unnat Jain and Vaishnavi Subramanian; friends from IIT Kanpur - Akash Gupta, Dhruv Yadav, Vipul Gupta, Sandesh Chopade and Shreshth Gandhi. I would also like to thank my friends from middle and high school in India as well as my wingmates from IIT Kanpur who have continued to support me over the years. I am quite sure that I have missed some names. I would like to preemptively apologize for any persons that I have missed.

I would also like to thank my collaborators - Prof. Atilla Eryilmaz, Prof. Bin Li, Prof. Lei Ying, Prof. Niao He, Prof. Weina Wang and Wentao Weng. I would also like to thank current and former CSL and ECE staff members who made my life a lot easier by streamlining several administrative tasks - Brenda Roy, Jamie Smith, Peggy Wells and Rachel Palmisano. I would like to extend my gratitude to Prof. Bruce Hajek, Prof. Maxim Raginsky and Prof. Niao He for agreeing to serve on my dissertation committee. I would also like to thank Jan Progen with the ECE Editorial Services for her help with proofreading my thesis.

Finally, I would like to thank Prof. R. Srikant for being an amazing PhD advisor, mentor and friend. As a PhD advisor, Srikant was respectful in his conduct, generous with his time and constructive with his feedback. As a mentor, he made sure that in addition to developing research acumen, I also learnt important non-technical ropes of professional life such as effective communication skills, cultivating healthy professional relationships etc. And as a friend, he made sure that I had an adequate work-life balance and that I could comfortably seek counsel from him regarding personal matters. I will always be grateful to Srikant for all the help that he extended to me over the years.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 STRUCTURED MULTI-ARMED BANDITS 4
2.1 Background and Problem Formulation 4
2.2 Existing Algorithms 8
2.3 CoTS: Constrained Thompson Sampling 14
2.4 Results 19

CHAPTER 3 BANDITS WITH TWO-LEVEL FEEDBACK 32
3.1 Motivation 32
3.2 Problem Formulation 34
3.3 Lower Bound on the Regret 37
3.4 Algorithm Design and Analysis 46
3.5 Experimental Results 54

CHAPTER 4 TWO TIME-SCALE TEMPORAL DIFFERENCE LEARNING ALGORITHMS 57
4.1 Background and Problem Formulation 57
4.2 Finite-Time Performance Bounds 61
4.3 Adaptive Selection of Learning Rates 65
4.4 Experiments 68
4.5 Supplementary Material 71

CHAPTER 5 DOUBLE Q-LEARNING ANALYSIS AND INSIGHTS 83
5.1 Introduction 83
5.2 Q-Learning and Double Q-Learning 85
5.3 Main Results 87
5.4 Numerical Results 92
5.5 Linearization Results 100
5.6 A Stronger Result for the Mean-Squared Error 108

CHAPTER 6 FUTURE DIRECTIONS 113

REFERENCES 114

CHAPTER 1

INTRODUCTION

Reinforcement learning (RL) has been instrumental in the recent advances made by artificial intelligence agents, especially in the domains of video games and board games (see [1], [2] and [3]). The success of reinforcement learning in these domains relies on the availability of vast amounts of simulated or real data that can be leveraged by a model-free learning algorithm to learn optimal policies. In addition to games, reinforcement learning has also been utilized gainfully in applications in robotics, self-driving vehicles and wireless networks. In many of these applications, data (simulated or real) is expensive to collect which makes popular model-free data-intensive reinforcement learning algorithms infeasible.

In contrast to model-free learning algorithms, model-based learning algorithms exploit the knowledge of the environment dynamics to learn the optimal policy, and are generally sample-efficient. If the model is exactly known, algorithms such as value iteration and policy iteration can be used (see [4]). If the model is approximately known or if fixed point iteration algorithms are computationally infeasible, model-based algorithms can still be of utility as the rollouts from the known dynamics can be used to train the policy, better estimate temporal difference targets, generate simulated data samples etc. (see [5], [6], [7], [8] and [9]). Although model-based algorithms are sample-efficient, highly complex environments such as those encountered in robotics or self-driving cars may not be amenable to such algorithms since defining the model in these applications is nearly impossible.

In this thesis, we will focus on model-free learning algorithms vis-à-vis sample-efficiency in four different reinforcement learning problems:

1. **Structured Multi-Armed Bandits:** We consider a multi-armed bandit (MAB) problem with additional structure which arises in several applications such as wireless network rate adaptation and the online pricing problem. Traditional MAB algorithms such as Thompson sam-

pling and KL-UCB can be used for this problem but these algorithms do not exploit the known additional structure. We will design and analyze an algorithm that exploits the additional structure in the problem resulting in a reduction in the data required to achieve performance comparable to the existing state-of-the-art. We will show, theoretically and in experiments, the efficacy of our proposed algorithm over traditional MAB algorithms. This work was presented at INFOCOM 2019 [10].

2. **Bandits with Two-Level Feedback:** Motivated primarily by the panoramic video streaming problem, we consider a stochastic multi-armed bandit problem with two-levels of feedback. Different from traditional bandit problems, in our problem, we receive two independent pieces of feedback information after each arm is selected. The overall reward associated with the arm is the product of these two pieces of information. In addition to the panoramic video streaming problem, this non-standard bandit setup also arises in other applications such as the web link selection problem. For this general problem, we propose a novel variant of the Kullback-Leibler Upper Confidence Bound (KL-UCB) algorithm, and show that it achieves asymptotically optimal regret, where regret measures the total expected loss in throughput compared to an oracle which has knowledge of the system statistics. We demonstrate the better performance of our algorithm as compared to the standard KL-UCB algorithm using synthetic and real datasets.

3. **Two Time-Scale Temporal Difference Learning Algorithms:** We analyze the finite-time performance of general two time-scale temporal difference learning algorithms with linear function approximation. Several popular reinforcement learning algorithms such as GTD, TDC etc. are special cases of our general formulation. Using the resulting finite-time performance bounds, we shed light on how different hyperparameters affect the rate of convergence of two time-scale RL algorithms, and subsequently present an adaptive learning rate selection rule which leads to much faster convergence as compared to traditional learning rate rules in our experiments. This work was presented at NeurIPS 2019 [11].

4. **Double-Q Learning Analysis and Insights:** We establish a theoretical comparison between the asymptotic mean-squared error of Double Q-learning and Q-learning. Our result builds upon an analysis for linear stochastic approximation based on Lyapunov equations which applies to both the tabular setting and with linear function approximation, provided that the optimal policy is unique and the algorithms converge. We show that the asymptotic mean-squared error of Double Q-learning is exactly equal to that of Q-learning if Double Q-learning uses twice the learning rate of Q-learning and outputs the average of its two estimators. We also present some practical implications of this theoretical observation using simulations. This work was presented at NeurIPS 2020 [12].

CHAPTER 2

STRUCTURED MULTI-ARMED BANDITS

2.1 Background and Problem Formulation

We will motivate the MAB problem with additional structure using the optimal link rate selection problem which arises in wireless networks. Note that the same problem formulation can be potentially used in other applications such as the online pricing problem.

Optimal link rate selection is an important problem especially in the context of 802.11 systems and other wireless networking systems (see [13], [14], [15] and [16]). At each time slot, the objective of the problem is to choose from a finite set of transmission rates to identify, as quickly as possible, the optimal rate, i.e., the rate maximizing the expected throughput. Along with 802.11 systems, the optimal link rate selection problem is also pertinent in cellular wireless systems, especially with the advent of mmWave technology. In this chapter, we consider a wireless network operating under some MAC protocol and focus on a particular link (transmitter-receiver pair) in this network. Time is indexed so that consecutive time slots are the time slots at which this link is chosen to transmit. Thus, we effectively consider a single link in our work and we are interested in choosing the optimal transmission rate for this link.

In particular, we consider a time varying wireless channel/link $(h(t))_{t\geq 0}$. At each time slot t, the channel allows transmission at one of the following n rates of transmission: $r_1, r_2, ..., r_n \in \mathcal{R}$. Without loss of generality, we assume $r_1 < r_2 < ... < r_n$. The corresponding probabilities of success for the transmission at these rates are assumed i.i.d. at each time slot and are given by the vector $\theta = (\theta_1, \theta_2, ..., \theta_n)$. Observe that, at a given time slot t, if a transmission at rate r will be successful in the particular channel state $h(t)$, transmission at all rates less than r will also be successful. Therefore,

$1 \geq \theta_1 \geq \theta_2 \geq ... \geq \theta_n \geq 0$. Let Θ denote the set of valid rate success probability vectors, i.e., $\Theta = \{\lambda : 1 \geq \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0\}$. The aim of optimal link rate selection is to transmit at the optimal rate r^* (at each time slot) so that the expected throughput is maximized. Let i^* be the index of r^* in the set $\mathcal{R}$, i.e., $r^* = r_{i^*}$. Mathematically, r^* essentially solves the following optimization problem:

$$r^* = r_{i^*} = \arg\max_{r_i} r_i \theta_i. \tag{2.1}$$

In order to understand the monotonicity of the components of θ, we further elaborate on the model by looking at the wireless channel in more detail. At any time t, the random channel $(h(t))_{t\geq 0}$ which we consider can be in one of the following n states: $h_1, h_2, ..., h_n$. Let $\mathcal{H} = \{h_1, h_2, ..., h_n\}$. Let the corresponding probabilities associated with these channel states be $\nu = (\nu_1, \nu_2, ..., \nu_n)$, i.e., $P\{h(t) = h_i\} = \nu_i, \forall 1 \leq i \leq n, \forall t \geq 0$. At each time slot t, the channel state $h(t)$ is drawn independently from the above distribution. Each channel state admits a maximum possible transmission rate, i.e., corresponding to each channel state $h_i \in \mathcal{H}$, we have a maximum possible rate r_i which can be successfully transmitted. Without loss of generality, we assume that $h_1, h_2, ..., h_n$ are ordered in the increasing order of their respective maximum admissible transmission rates, i.e., $r_1 < r_2 < ... < r_n$. As before, let $\mathcal{R} = \{r_1, r_2, ..., r_n\}$. Note that if the channel is in state h_k, it can admit transmission rates $r_i, 1 \leq i \leq k$. Therefore, for any rate r_i the probability of being successfully transmitted at any time t is $\sum_{j=i}^{n} \nu_j$. From the definition of θ_i, we have $\theta_i = \sum_{j=i}^{n} \nu_j$, resulting in the monotonic structure.

If the vector θ is known, the optimization problem in Equation (2.1) can be solved easily. But, in most practical applications, the channel statistics are unknown and hence there is no information on the vector θ. This lack of information necessitates the use of online/sequential learning algorithms, which learn the optimal rate over time by transmitting at various rates and gaining information about their probabilities of success (from the history of transmissions and their outcomes). Such online algorithms encounter an *exploration vs. exploitation* trade-off (see [17], a survey on multi-armed bandit problems), i.e., while they have to explore different rates to gain more accurate information, they also have to simultaneously exploit the information

gained to transmit at the best possible rate.

A quantity often used to quantify the performance of online algorithms is expected regret. In order to define expected regret, we will introduce some notation first. Since the model we use is similar to the one used in [13] and [15], we will use similar notation to make our analysis and results more accessible to a reader familiar with those works. Let $r(t)$ denote the rate of transmission chosen at time slot t. Let $i(t)$ denote the index of $r(t)$ in the set of rates $\mathcal{R}$, i.e., $r(t) = r_{i(t)}$. Let $X(t)$ denote the outcome of the transmission at time slot t, i.e., $X(t) = 1$ in the case of a successful transmission and $X(t) = 0$ otherwise. Note that $X(t)$ is a Bernoulli random variable with parameter $\theta_{i(t)}$. Observe that the optimization problem given by Equation (2.1) can be rewritten as:

$$r^* = r_{i^*} = \arg\max_{r_i} \mathbb{E}[r(t) \times X(t) | r(t) = r_i, \theta].$$

Expected regret for T time slots is defined as the expected loss in throughput incurred by the algorithm due to transmission at sub-optimal rates. Let $R(T)$ denote the regret for T time slots. Mathematically:

$$\mathbb{E}[R(T)] = \mathbb{E}[\sum_{i=1}^{T} \{r_{i^*}\theta_{i^*} - r_{i(t)}\theta_{i(t)}\}].$$

Let $N_i(T)$ denote the number of times transmission was made at rate r_i until time T. Also, let $\Delta_i = r_{i^*}\theta_{i^*} - r_i\theta_i$ denote the loss in expected throughput because of transmitting at rate r_i instead of rate r_{i^*}. A more useful way to write expected regret is the following:

$$\mathbb{E}[R(T)] = \mathbb{E}[\sum_{i \neq i^*} N_i(T)\Delta_i] = \sum_{i \neq i^*} \mathbb{E}[N_i(T)]\Delta_i. \tag{2.2}$$

Another quantity which is useful in quantifying the performance of online algorithms is simply the number of times transmissions at sub-optimal rates are made, i.e.:

$$R'(T) = \sum_{i \neq i^*} N_i(T). \tag{2.3}$$

Note that $R'(T)$ is a random variable. We will study both the expected

regret and $R'(T)$ in this chapter.

In [13], the authors tackle the optimal link rate selection problem by treating each rate as an independent arm in the standard multi-armed bandit problem setup. Although this approach overcomes certain challenges associated with the problem, it does not exploit the structure in the set Θ as treating the rates as independent arms implies lack of ordering in the components of the vector θ. They take a KL-UCB inspired frequentist approach and present an algorithm called KL-R-UCB, which achieves logarithmic regret. With an additional assumption that the expected throughput at different rates is unimodal, they present an asymptotically optimal algorithm called G-ORS (also, see [18] for a Thompson sampling inspired algorithm for the unimodal case).

In [15], the authors treat the problem similarly and present a Thompson sampling inspired algorithm called Modified Thompson Sampling (MTS). This algorithm takes a Bayesian approach and is shown to have the same regret upper bound as KL-R-UCB, since it also does not exploit the structure in the set Θ. They also provide a lower bound for the problem but only for the very specific case of three channel states and rate $r_1 = 0$. It is worth noting that while both KL-R-UCB and MTS have been shown to have the same regret upper bound, simulations in this chapter indicate that MTS performs significantly better than KL-R-UCB. Also, in our work, we do not need any additional assumptions on Θ such as unimodality since such assumptions are hard to justify in practice. However, our algorithm can easily incorporate any additional structure including unimodality.

Our main contributions are the following:

1. We have designed an algorithm called Constrained Thompson Sampling (CoTS). CoTS exploits the structure in Θ efficiently (i.e., the fact that $\theta_1 \geq \theta_2 \geq ... \geq \theta_n$) and is more general in the sense that any additional structure in Θ (such as unimodality) can also be incorporated with minor tweaks in the same algorithm (unlike previous approaches where different constraints were tackled using very different algorithms). We also present SITS, an approximate but efficient way to implement CoTS in practice (see Sections 2.3.1 and 2.3.2).

2. We provide theoretical guarantees for the regret achieved by CoTS by proving a high probability large-horizon logarithmic upper bound for

the notion of regret quantified by $R'(T)$ (see Section 2.4.1).

3. We prove an asymptotic lower bound for the expected regret (defined in Equation (2.2)) achieved by any algorithm for the optimal link rate selection problem. We note that this lower bound is established without any unimodality assumption as in [13] or any assumptions on the number of channel states or rates as in [15] (see Section 2.4.2).

4. We provide numerical results to establish the superiority of CoTS over the current state-of-the-art algorithms and to show that it achieves the theoretical lower bound (see Section 2.4.3).

2.2 Existing Algorithms

In this section we discuss some existing work on the optimal link rate selection problem, before moving on to the next section where we present CoTS.

2.2.1 Sampling and Measurement-based Algorithms

Several link rate selection algorithms (also known as rate adaptation algorithms) relying on sampling-based approaches have been proposed in the literature (for example, see [19], [20] and [21]). At any time slot, these methods rely on the history of outcomes for transmissions at different available rates to determine the optimal rate to transmit at. These algorithms primarily use well-engineered heuristics to strike a balance between exploration and exploitation.

Another class of algorithms which can potentially be used are the ones which rely on measurements quantifying the quality of the channel (for example, see [22], [23] and [24]). If the measurements obtained are accurate then these methods can perform really well, but in several practical scenarios that arise in modern time-varying wireless systems, it is costly to obtain reliable measurements. Hence, the viability of such measurement-based algorithms is unclear.

2.2.2 Multi-Armed Bandits-based Algorithms

In order to tackle the exploration vs. exploitation trade-off intrinsic to the link rate selection problem in a *theoretically principled* and *optimal* manner, a number of recent papers have focused on solving the link rate selection problem using algorithms from the multi-armed bandits and stochastic optimization literature (see [13] and [15]). In the standard stochastic multi-armed bandit problem, we have several actions (or arms) available to us and at every time slot, we need to choose one of the available actions to play. Once an action is played, we receive a random reward. Corresponding to every action, the random reward is drawn from a probability distribution with a finite expected value. The reward for the action played is independent and identically distributed (i.i.d.) at every time slot.

The objective of the problem is to design an algorithm that determines the best action to play at any time slot, i.e., the action with the maximum expected value of the reward outcome. The algorithm has access to the history of actions played and the corresponding random reward outcomes until the latest time slot and can use this history to choose the next action. The multi-armed bandit problem is a well-studied problem in literature (see [17] for a survey). We will now discuss two existing algorithms in this category and elaborate on their advantages and shortcomings.

KL-UCB-based

In [13], the authors present a KL-UCB (a variant of the classical UCB algorithm, see [25] and [26]) inspired algorithm called KL-R-UCB (see Algorithm 1). In KL-R-UCB, at each time slot t, the algorithm computes an index $q_i(t), \forall r_i$ as follows:

$$q_i(t) = \max\{q \in [0, r_i] : n_i(t) D(\frac{\hat{\mu}_i(t)}{r_i}, \frac{q}{r_i}) \leq \log(t) + c \log\log(t)\},$$

where $n_i(t)$ denotes the number of times rate r_i has been transmitted in t time slots, $\hat{\mu}_i(t)$ denotes the empirical average of all the reward outcomes of those transmissions and $D(x, y)$ denotes the KL divergence between two Bernoulli distributions parametrized by x and y. It is shown in [13] that KL-R-UCB achieves logarithmic regret although it does not exploit the structure in Θ.

Making an additional assumption of unimodality of expected throughput, i.e., $r_1\theta_1 \leq r_2\theta_2 \leq ... < r_{i^*}\theta_{i^*} > r_{i^*+1}\theta_{i^*+1} \geq ... \geq r_n\theta_n$, authors in [13] present another KL-UCB inspired algorithm called G-ORS (also see [18] for a similar Thompson sampling inspired algorithm) which is very different from KL-R-UCB and is asymptotically optimal.

Algorithm 1 KL-R-UCB algorithm

---

for $t = 1, 2, \ldots, n$: transmit at rate r_t.

for $t = n+1, n+2, \ldots$:

1. Compute the set $\mathcal{I} = \arg\max_i q_i(t)$.
2. Transmit at rate $r_{i(t)}$ where $i(t) \in \mathcal{I}$.

end for

Thompson Sampling-based

Thompson sampling is a popular algorithm that is applied to solve the multi-armed bandit problem. In [27], Agrawal and Goyal obtain an upper bound on the regret (expected reward loss because of the non-optimal actions played) due to standard Thompson sampling for Bernoulli (see Algorithm 2) as well as non-Bernoulli rewards, and show that it matches a lower bound due to Lai and Robbins (see [28]) in the asymptotic regime (when the number of times the bandit/game is played approaches infinity).

As the authors in [15] observe, the optimal link rate selection problem falls in the more general problem setup considered in [29]. Therefore, in principle, one can use the general Thompson sampling algorithm (Algorithm 3) for the problem we consider. However, a direct implementation is infeasible as we discuss next. Also note that, while there are no known lower bounds in the more general multi-armed bandit settings, it has been shown in [29] that the regret is still upper bounded logarithmically as a function of time T.

Algorithm 2 Thompson sampling for Bernoulli rewards

for each action $a_i, i = 1, 2, ..., n$, set $S_i = 0$ and $F_i = 0$.
for each $t = 1, 2, \ldots :$

1. For all actions a_i, draw $\nu_i(t) \sim$Beta$(S_i + 1, F_i + 1)$.[1]
2. Play action $a_{i(t)}$, where $i(t) = \arg\max_i \nu_i(t)$.
3. Observe the random reward $g(t)$.
4. (Posterior Update for Prior) If $g(t) = 1$, set $S_{i(t)} = S_{i(t)} + 1$. Else if $g(t) = 0$, set $F_{i(t)} = F_{i(t)} + 1$.

end for

Algorithm 3 General Thompson sampling

initialize prior $p_\nu(1)$ (for channel state probability vector ν).
for each $t = 1, 2, \ldots :$

1. Draw $\nu(t) \sim p_\nu(t)$. Compute $[\theta(t)]_i = \sum_{j=i}^{n} [\nu(t)]_j$.
2. Transmit at rate $r_{i(t)}$, where $r_{i(t)} = r_{opt}(\theta(t))$.
3. Observe the random transmission outcome $X(t)$.
4. (Prior Update) Set $p_\nu(t+1) \propto \mathbb{P}(X(t)|\nu) p_\nu(t)$.

end for

Challenges

The major challenges which arise if we use Algorithm 3 for our problem are as follows:

1. While dealing with the rate admissibility probabilities θ, it is difficult to come up with a feasible prior distribution $(p_\theta(t))$ for implementing

[1] Beta(a, b), known as the beta distribution is a continuous probability distribution with pdf: $f_{a,b}(x) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)}, x \in [0, 1]$, $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

the general Thompson sampling algorithm. Since the rate admissibility probability distribution is not multinomial and has interdependent components, the prior required would be complicated and difficult to update. However, one can use Thompson sampling to estimate the channel state probability ν (Algorithm 3), but it comes at a huge computational cost as we discuss next.

2. If we deal with the multinomial channel state distribution ν, we can use the popular Dirichlet distribution as the prior over $\mathcal{V}$. But since we observe only the outcome of our transmission and not the exact channel state, the posterior update for the Dirichlet prior distribution may require exponentially increasing storage and computational power depending on the trajectory of the algorithm. For example, let us consider the case where $n = 3$, i.e., there are three possible states the channel can take. At $t = 1$, we start with a Dirichlet distribution as prior with parameters $(1, 1, 1)$, i.e., Dir$(1, 1, 1)$. Suppose at $t = 1$, we transmit at rate r_2 and it is successful. We simply know that the channel is either in channel state 2 or 3. Therefore, after standard calculations, the prior becomes:

$$\frac{B(1,2,1)}{B(1,2,1)+B(1,1,2)}Dir(1,2,1)+\frac{B(1,1,2)}{B(1,2,1)+B(1,1,2)}Dir(1,1,2),$$

where $B(\alpha) = \frac{\prod_{i=1}^{k}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k}\alpha_i)}$ and $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$. Clearly, we now need to store two sets of Dirichlet parameters instead of one. As the number of iterations increase, the number of parameters to be stored and evaluated increases exponentially. After t time slots, the number of Dirichlet distribution parameters to be stored and evaluated could be as high as 2^t. This renders the algorithm infeasible due to memory and computational constraints.

Due to the infeasibility of Algorithm 3, in [15], the authors use Algorithm 2 and use it to design a new link rate selection algorithm which they call Modified Thompson Sampling (MTS) algorithm. At any time slot t, MTS maintains independent beta priors for every individual component of θ, then samples a vector $\lambda(t)$ from the product of these priors and transmits at the rate optimal for the sampled vector (see Algorithm 4). Finally, depending on the outcome, it does a Bayesian update to the prior corresponding to

the component of θ having the same index as that of the rate at which the transmission was made. Since MTS considers independent beta priors for every component of θ, the set of valid parameters it explores is $[0,1]^n$ instead of $\Theta = \{x \in [0,1]^n : x_1 \geq x_2 \geq ... \geq x_n\}$. Therefore, it is not an exact Thompson sampling algorithm, but rather a Thompson sampling-based algorithm. It is shown in [15] that MTS also achieves logarithmic regret, similar to KL-R-UCB, since KL-R-UCB also does not exploit the fact that the components of θ are non-increasing. However, as shown in the experimental results section later, MTS seems to perform better than KL-R-UCB in simulations.

Algorithm 4 Modified Thompson sampling algorithm

---

for each rate $r_i, i = 1, 2, ..., n$, set $s_i = 0$ and $f_i = 0$.
for $t = 1, 2, \ldots :$

1. For every rate r_i, draw $\lambda_i(t) \sim \text{Beta}(s_i + 1, f_i + 1)$.
2. Compute $i(t) = \arg\max_i r_i \lambda_i(t)$. Transmit at rate $r_{i(t)}$.
3. Observe the random transmission outcome $X(t)$.
4. (Prior Update) If $X(t) = 1$, set $s_{i(t)} = s_{i(t)} + 1$. Else if $X(t) = 0$, set $f_{i(t)} = f_{i(t)} + 1$.

end for

2.2.3 Areas of Improvement

From the above discussion, we observe that there are two major disadvantages associated with the current state-of-the-art MAB-based algorithms for the optimal link rate selection problem:

1. The current state-of-the-art algorithms such as KL-R-UCB and MTS do not exploit the basic structure in the set Θ, i.e., they do not take advantage of the fact that the probability of success is a non-increasing function of the rate of transmission. If an algorithm can exploit this structure in the problem, it can potentially outperform both KL-R-UCB and MTS.

2. Additional constraints or structure in the set Θ (such as unimodality of the expected throughput) are not handled easily by the current state-of-the-art algorithms. In fact, even for unimodality, there is a completely different set of algorithms. If an algorithm can handle additional constraints more generally, it will be useful in a much wider set of applications and environments.

In Section 2.3, we will present CoTS which overcomes the above mentioned disadvantages. CoTS uses the basic structure in Θ to its advantage and at the same time is amenable to several additional constraints in Θ that one might want to incorporate.

2.3 CoTS: Constrained Thompson Sampling

The reason why KL-R-UCB and MTS do not perform optimally is because they do not exploit the basic structure in the set Θ. Moreover, with additional structure in the set such as unimodality, the performance of these algorithms deteriorates further and one has to come up with different algorithms which are optimal. In the context of these observations, we now present CoTS (see Algorithm 5) and the intuition behind it.

2.3.1 Intuition

The idea behind CoTS is intuitive and simple. At each time slot t, we maintain independent beta priors for each component of θ, similar to MTS. But instead of simply sampling from the product of these priors (as in MTS), we sample from a distribution which is proportional to the product of these priors when the value being sampled, say λ, belongs to Θ and is 0 otherwise. Mathematically, we sample from a distribution with the following p.d.f.:

$$p_t(\lambda) \propto \mathbb{1}\{\lambda \in \Theta\} \prod_{i=1}^{n} \text{Beta}(s_i(t)+1, f_i(t)+1)(\lambda_i),$$

where $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n) \in [0,1]^n$, and $s_i(t)$ and $f_i(t)$ are the number of successful and failed transmissions respectively until the beginning of the time slot t, for the rate r_i. This simple modification allows us to exploit

the structure in Θ by assigning non-zero probability only to the parameters which belong to it.

Algorithm 5 Constrained Thompson sampling algorithm

---

for each rate $r_i, i = 1, 2, ..., n$, set $s_i = 0$ and $f_i = 0$.
for $t = 1, 2, \ldots$:

1. Draw $\lambda(t) \sim \mathbb{1}\{\lambda(t) \in \Theta\} \times \prod_{i=1}^{n} \text{Beta}(s_i + 1, f_i + 1)$.
2. Compute $i(t) = \arg\max_i r_i \lambda_i(t)$. Transmit at rate $r_{i(t)}$.
3. Observe the random transmission outcome $X(t)$.
4. (Prior Update) If $X(t) = 1$, set $s_{i(t)} = s_{i(t)} + 1$. Else if $X(t) = 0$, set $f_{i(t)} = f_{i(t)} + 1$.

end for

In [15], the authors state that the reason one has to treat different components of θ independently is that it is difficult to come up with an easy-to-update prior for Thompson sampling that incorporates the non-increasing property of the components of θ (or any other structure such as unimodality). In CoTS, we still maintain different beta priors for each component of θ to keep the updates simple, but in order to exploit the structure in Θ, we restrict the joint distribution to have non-zero weight only for valid parameters in Θ. At any time slot t, let the rate selected for transmission be $r_{i(t)}$ and let the outcome of transmission be $X(t)$. Then, the prior $p_{t+1}(\lambda)$ after the Bayesian update will be:

$$p_{t+1}(\lambda) \propto \mathbb{1}\{\lambda \in \Theta\} \prod_{i=1}^{n} \text{Beta}(s_i(t) + 1, f_i(t) + 1)(\lambda_i) \times \lambda_{i(t)}^{X(t)} (1 - \lambda_{i(t)})^{1 - X(t)}.$$

Simplifying the above expression, we get:

$$\begin{aligned} p_{t+1}(\lambda) \propto \mathbb{1}\{\lambda \in \Theta\} & \prod_{i \neq i(t)} \text{Beta}(s_i(t) + 1, f_i(t) + 1)(\lambda_i) \\ & \times \text{Beta}(s_{i(t)}(t) + 1 + X(t), f_{i(t)}(t) + 1 + (1 - X(t))). \end{aligned} \tag{2.4}$$

Hence, to update the prior distribution, we need to simply update the number

of successes or failures corresponding to the beta prior of the component of θ with the same index as that of the rate transmitted (Step 4 in Algorithm 5). Thus, maintaining different beta priors for every component of θ allows CoTS to have easy prior updates (similar to MTS), whereas the restriction imposed on the joint distribution allows it to exploit the structure in Θ. Observe that CoTS is essentially an exact Thompson sampling algorithm, whereas MTS is not. Therefore, a different prior distribution can also be used in place of the prior distribution used by CoTS as long as its Bayesian update is easy and exact.

Also, note that CoTS is general in the sense that the set Θ can have any additional structure on top of the basic property of non-increasing components (such as unimodality) and the algorithm will still work. The indicator function in the joint prior distribution can incorporate any structure in the set Θ, while keeping the prior updates simple as shown in Equation (2.4). Therefore, CoTS allows us to overcome both the disadvantages associated with the current state-of-the-art MAB-based algorithms discussed in the previous section.

2.3.2 Efficient Implementation

In this section, we will discuss some efficient ways of implementing CoTS. Since the prior update step is straightforward, the main focus for improving the efficiency lies on Step 1, i.e., sampling $\lambda(t)$ from the prior distribution.

Rejection Sampling

One straightforward way to implement CoTS is to use rejection sampling, i.e., sample $\lambda(t) \sim \prod_{i=1}^{n} \text{Beta}(s_i + 1, f_i + 1)$ and reject the samples until $\lambda(t) \in \Theta$. The main advantage of rejection sampling is that it is easy to implement. Also, rejection sampling is general in the sense that as long as the operation of checking whether a sampled value lies in Θ can be done efficiently, it does not require any other problem-dependent alterations. But the main disadvantage of rejection sampling is that it can be really slow. For example, if the probability of obtaining a valid parameter $\lambda(t) \in \Theta$ when sampling from the distribution $\prod_{i=1}^{n} \text{Beta}(s_i+1, f_i+1)$ is x, then the expected

number of times in which one samples a valid parameter is $\frac{1}{x}$. Thus, if x is really small, the expected sampling time is really large. Therefore, we need to have a faster sampling method, especially in the cases where the progress of the algorithm will result in x taking small values.

Sequential Inverse Transform Sampling (SITS)

For the basic structure in Θ, as well as for unimodality, we present a technique to speed up the sampling step for CoTS, called Sequential Inverse Transform Sampling (SITS). The idea behind SITS is to sample different components of $\lambda(t)$ sequentially (instead of all at once and then rejecting), while simultaneously ensuring that the sampled components satisfy the structure in Θ. Note that due to normalization issues, this technique will not exactly sample from the CoTS distribution. But, as the algorithm progresses, SITS becomes increasingly accurate. We will illustrate the idea below for the non-increasing structure in Θ as well as for the additional structure of unimodality.

Non-Increasing Structure:

Consider the basic non-increasing components structure in Θ. We observe that the prior distribution at time t can be written as:

$$p_t(\lambda) \propto \prod_{i=1}^{n} \mathbb{1}\{\lambda_{i-1} \geq \lambda_i\} \text{Beta}(s_i + 1, f_i + 1),$$

where $\lambda_0 = 1$. Therefore, to sample fast, we can use the following heuristic: sample $\lambda_1(t) \sim \text{Beta}(s_1 + 1, f_1 + 1)$, then sample $\lambda_2(t) \sim \text{Beta}(s_2 + 1, f_2 + 1)$ while restricting it to be less than $\lambda_1(t)$ and so on. To sample a random variable Z from $\text{Beta}(x, y)$ quickly while restricting it to lie between interval $[a, b]$ (instead of interval $[0, 1]$), we can use the standard inverse transform sampling as follows:

1. Let F denote the cumulative distribution function of $\text{Beta}(x, y)$. Let $\alpha_0 = F(a)$ and $\alpha_1 = F(b)$.

2. Sample a random variable U uniformly from the interval $[\alpha_0, \alpha_1]$, i.e., $U \sim \mathcal{U}(\alpha_0, \alpha_1)$.

3. $Z = F^{-1}(U)$ is the required random variable.

The above technique speeds up the sampling process and unlike rejection sampling, makes the sampling time independent of the probability of sampling a valid parameter from $\prod_{i=1}^{n}\text{Beta}(s_i+1, f_i+1)$.

Unimodality:

For the case of unimodality, a similar procedure can be followed as in the non-increasing case except that now for every component being sampled, we need to ensure that it continues to maintain unimodality along with the non-increasing property. That can be achieved as follows:

1. At any time t, first sample $\lambda_1(t) \sim p_t(\lambda_1) = \text{Beta}(s_1+1, f_1+1)$.
2. Subsequently, sample $\lambda_2(t) \sim p_t(\lambda_2) \propto \mathbb{1}\{\lambda_1(t) \geq \lambda_2(t)\}\text{Beta}(s_2+1, f_2+1)$.
3. For $i = 3, 4, ..., n$:

   Sample $\lambda_i(t) \sim p_t(\lambda_i) \propto \mathbb{1}\big\{\min\{\lambda_{i-1}(t), \rho\} \geq \lambda_i(t)\big\}\text{Beta}(s_i+1, f_i+1)$, where ρ is the solution to the following equation:

$$\begin{aligned} &\max_{0\leq x\leq 1} x \\ &\text{subject to:} \\ &\min\{r_{i-2}\lambda_{i-2}(t), r_{i-1}\lambda_{i-1}(t), r_i x\} \neq r_{i-1}\lambda_{i-1}(t). \end{aligned} \tag{2.5}$$

Note that sampling from the truncated Beta distribution in steps 2 and 3 above is the same as truncated Beta sampling explained in steps 1 to 3 in the non-increasing structure case.

The intuition behind the above procedure is simple. Recall that unimodality implies the following: $r_1\theta_1 \leq r_2\theta_2 \leq ... < r_{i^*}\theta_{i^*} > r_{i^*+1}\theta_{i^*+1} \geq ... \geq r_n\theta_n$. This essentially implies that there can be no local minimum in the throughput vector (see Fig. 2.1 for an illustration). Restricting the component being sampled to be less than or equal to ρ (obtained by solving the optimization problem in Equation (2.5)) ensures that no local minimum is created while sampling different θ components.

Remark 1. *To make SITS as accurate as possible, we sample the different components of λ in the increasing order of their variance. The intuition behind that is to sample the less "random" component first so that the normalization error is not very high. Also, note that the computational and*

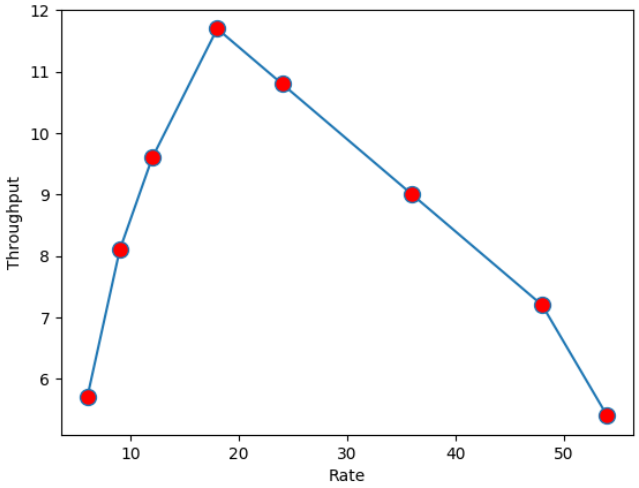


Figure 2.1: Illustration of unimodality.

storage complexity of CoTS (exploiting the basic structure of non-increasing components) is linear in the number of rates, same as that of KL-R-UCB and MTS.

2.4 Results

2.4.1 Upper Bound

In this section, we present theoretical guarantees for the performance of CoTS in terms of a high probability large-horizon upper bound on the number of times transmissions at sub-optimal rates are made. We utilize the results obtained in [29] to this end. As in [29], we make some simplifying assumptions to make the analysis tractable. In particular, we assume that the possible values for Θ lie in a discrete set. We state the assumptions more precisely next.

Let π_t denote the prior at the beginning of time slot t. We make the following assumptions:

Assumption 1 (Finitely many transmission rates). $|\mathcal{R}| < \infty$.
Assumption 2 (Finite Θ and non-zero initial probability on θ). $|\Theta| < \infty$, i.e, $\Theta = \{\zeta^{(1)}, \zeta^{(2)}, ..., \zeta^{(L)}\}$. Moreover, $\theta \in \Theta$ and $\pi_1(\theta) > 0$.

Assumption 3 (Strictly decreasing probability of success). For all $\zeta \in \Theta$, $\zeta_1 > \zeta_2 > ... > \zeta_n$.

Assumption 4 (Unique optimal rate) The optimal transmission rate is unique, i.e., $r_{i^*}\theta_{i^*} > r_i\theta_i, \forall i \neq i^*$.

Under the above assumptions, we have the following result:

Theorem 1. *Let $N_i(T)$ denote the number of times a transmission at rate r_i is made until time slot t. Under Assumptions 1-4, a high probability large-horizon upper bound holds for CoTS as follows. For any $\delta, \epsilon \in (0,1)$, $\exists$ $T^* \geq 0$, such that $\forall$ $T \geq T^*$, with probability at least $1-\delta$, we have:*

$$R'(T) = \sum_{i \neq i^*} N_i(T) \leq \left(\frac{1+\epsilon}{1-\epsilon}\right) \sum_{i \neq i^*} \frac{\log T}{D(\theta_i, \frac{r_{i^*}\theta_{i^*}}{r_i})} + C,$$

where $C = C(\delta, \epsilon, \mathcal{R}, \Theta, \pi)$ is a problem-dependent constant independent of T and $D(x,y)$ denotes the KL divergence between two Bernoulli distributions parametrized by x and y respectively.

Proof. Our upper bound analysis uses the main result from [29], which gives a high probability large-horizon upper bound for the number of times a sub-optimal action is played by exact Thompson sampling for a complex online problem. As discussed in the previous section, CoTS is an exact Thompson sampling algorithm for the optimal link rate selection problem and hence the main result from [29] can be used to quantify its performance.

We note that the optimal link rate selection problem is a special case of the general complex online problem setup outlined in [29]. The set of actions $\mathcal{A}$ we have is essentially the set of transmission rates, i.e. $\mathcal{A} = \mathcal{R}$. Also, the observation space is $\mathcal{Y} = \{0, r_1, r_2, ..., r_n\}$, i.e., $\mathcal{Y}$ is the sample space for the possible rewards in terms of throughput. The reward function $h : \mathcal{Y} \to \mathbb{R}$ is the identity function, i.e., reward $z = h(y) = y$. Let $l(y; i, \theta)$ denote the probability of observing $y \in \mathcal{Y}$ when a transmission at rate r_i is made, with the underlying rate success vector θ. For all $y \in \mathcal{Y}$, we have $l(y; i, \theta)$ as follows:

$$l(y; i, \theta) = \begin{cases} \theta_i, & \text{if } y = r_i, \\ 1 - \theta_i, & \text{if } y = 0. \end{cases}$$

Hence, the optimal link rate selection problem is a special case of the gen-

eral complex online problems considered in [29]. Therefore, the above observation, along with Assumptions 1-4 and the fact that CoTS is an exact Thompson sampling algorithm imply that we can use Theorem 1 from [29] to quantify the performance of CoTS.

Using Theorem 1 from [29], $\forall\ T \geq T^*,$ for any $\delta, \epsilon \in (0,1)$, with probability at least $1-\delta$,

$$\sum_{i \neq i^*} N_i(T) \leq B(\log T) + C'(\delta, \epsilon, \mathcal{R}, \Theta, \pi), \tag{2.6}$$

where $C'(\delta, \epsilon, \mathcal{R}, \Theta, \pi)$ is a problem dependent constant and $B(\log T)$ is given as follows:

$$\begin{aligned}
&B(\log T) \coloneqq \\
&\max \sum_{i=1}^{n-1} z_i(a_i) \\
&\text{s. t. } z_i \in \mathbb{Z}_+^{n-1} \times \{0\}, a_i \in \mathcal{R} \setminus \{r_{i^*}\}, \\
&\quad z_k \succeq z_i, z_k(a_i) = z_i(a_i), k \geq i, \\
&\quad \forall 1 \leq j, i \leq n-1 : \\
&\quad\quad \min_{\lambda \in S_{a_i}(\theta)} \sum_{k=1}^{n-1} z_i(a_k) D(\theta_k, \lambda_k) \geq \frac{1+\epsilon}{1-\epsilon} \log T, \\
&\quad\quad \min_{\lambda \in S_{a_i}(\theta)} \sum_{k=1}^{n-1} (z_i(a_k) - \mathbb{1}_{\{k=j\}}) D(\theta_k, \lambda_k) < \frac{1+\epsilon}{1-\epsilon} \log T,
\end{aligned} \tag{2.7}$$

where $S_{a_i}(\theta)$ is the set of $\lambda \in \Theta$ which are indistinguishable from θ when r_{i^*} is transmitted and for which a_i is the optimal rate of transmission, i.e.:

$$S_{a_i}(\theta) \triangleq \{\lambda \in \Theta : D(\theta_{i^*}, \lambda_{i^*}) = 0 \text{ and } \arg\max_{r_k} r_k \lambda_k = a_i\}.$$

The interpretation of the optimization problem given by Equation (2.7) is as follows: $\{a_k\}_{k=1}^{n-1}$ is the sequence in which the sub-optimal rates are eliminated by CoTS, i.e., first the rate a_1 is eliminated, then the rate a_2 is eliminated and so on. z_i is the vector storing the number of times transmissions at sub-optimal rates have been made, until the time slot when rate a_i is eliminated. Once a rate is eliminated, it is not transmitted again.

Let $h(i)$ denote the index of the rate a_i in the set $\mathcal{R}$, i.e., $a_i = r_{h(i)}$. Now, we

will show that regardless of the sequence in which the rates are eliminated, for a rate $a_i = r_{h(i)}$, any feasible z_i should satisfy $z_i(a_i) \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{\log T}{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})} + 1$. Let us assume on the contrary that $z_i(a_i) > \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{\log T}{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})} + 1$. We will show that z_i cannot a feasible point of the optimization problem in Equation (2.7) because it violates the last constraint. For any $\lambda \in S_{a_i}(\theta)$, $\lambda_{i^*} = \theta_{i^*}$ and $\lambda_{h(i)} \geq \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}} > \theta_{h(i)}$. We have, for $j = i$:

$$\begin{aligned}
\sum_{k=1}^{n-1}(z_i(a_k) - \mathbb{1}_{\{k=i\}})D(\theta_k, \lambda_k) &\geq (z_i(a_i) - 1)D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}}) \\
&> \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})}{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})}\log T \\
&> \left(\frac{1+\epsilon}{1-\epsilon}\right)\log T.
\end{aligned}$$

The first inequality follows from the non-negativity of $z_i(a_k), \forall k$ and the fact that $D(x, y) \geq 0, \forall x, y$. The second inequality follows from the assumption $z_i(a_i) > \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{\log T}{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})} + 1$. Therefore, $z_i(a_i) > \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{\log T}{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})} + 1$ cannot be a feasible point of Equation (2.7) as it violates the last constraint. Hence, $z_i(a_i) \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{\log T}{D(\theta_{h(i)}, \frac{r_{i^*}\theta_{i^*}}{r_{h(i)}})} + 1$. Therefore,

$$\sum_{i=1}^{n-1} z_i(a_i) \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)\sum_{i \neq i^*}\frac{\log T}{D(\theta_i, \frac{r_{i^*}\theta_{i^*}}{r_i})} + n - 1.$$

Combining the above inequality with Equations (2.6) and (2.7), we get the result. □

2.4.2 Lower Bound

In [15], the authors prove a lower bound for the optimal link rate selection problem using a Lai and Robbins style of analysis (see [28]), but only in the special case of three channel states and rate $r_1 = 0$. In this section, we obtain a general lower bound for the problem, i.e., a lower bound obtained without any assumptions on the number of channel states or the rates.

In order to obtain the general lower bound, we transform the optimal

link rate selection problem setup into a controlled Markov chain framework (similar to [13]) and use results from [30] (quantifying the performance of efficient adaptive decision rules in a controlled Markov chain setup). The result is the following:

Theorem 2. *Let $P = \{i_1, i_1 + 1, ..., i^*, ..., n\}$ denote the set of indices such that for any $i \in P$, $r_i \geq r_{i^*}\theta_{i^*}$. Let $P' = P \setminus \{i^*\}$. Then, for the n-rates optimal link rate selection problem, the lower bound on expected regret (asymptotically) is given by:*

$$\lim_{T\to\infty} \frac{\mathbb{E}[R(T)]}{\log T} \geq \sum_i c_i \Delta_i, i \neq i^*,$$

where $\Delta_i = r_{i^*}\theta_{i^*} - r_i\theta_i$. The constants c_i are defined as follows:

$\forall\ i \in \{1, 2, ..., i^* - 1\}$, the constants c_i are the solution to the following linear program:

$$\begin{aligned} &\min \sum_{i=1}^{i^*-1} c_i(r_{i^*}\theta_{i^*} - r_i\theta_i), \\ s.\ t.\ &\sum_{l=1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}) \geq 1, \forall i \in P', \\ &c_i \geq 0, \forall i, \end{aligned} \tag{2.8}$$

$\forall\ i \in \{i^* + 1, i^* + 2, ..., n\}$, the constants c_i are the solution to the following linear program:

$$\begin{aligned} &\min \sum_{i=i^*+1}^{n} c_i(r_{i^*}\theta_{i^*} - r_i\theta_i), \\ s.\ t.\ &\sum_{l=i^*+1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}) \geq 1, \forall i, \\ &c_i \geq 0, \forall i, \end{aligned} \tag{2.9}$$

where $D(x, y)$ denotes the KL divergence between two Bernoulli distributions parametrized by x and y respectively.

Proof. Our lower bound analysis uses results obtained in [30] which quantify the performance of efficient adaptive decision rules in a controlled Markov

chain framework. In order to use these results, we need to transform our problem to a controlled Markov chain framework. We use the same transformation as used in [13]. For improving the readability of the users already familiar with the aforementioned references, we will reproduce the transformation from [13] and use similar notation as found in [30] and [13].

Consider a controlled Markov chain $(X_t)_{t\geq 0}$ on a finite state space $\mathbb{S} = \{0, r_1, r_2, ..., r_n\}$ with control laws given by the set $\mathbb{U} = \{1, 2, ..., n\}$. The control laws are independent of the state of the Markov chain and correspond to the index of the rate of transmission selected, i.e., if the control law i is selected, the same control (selecting rate r_i) is applied regardless of the state of the Markov chain. Let the transition probability for going from any state $x \in \mathbb{S}$ to any state $y \in \mathbb{S}$ be denoted by $p(x, y; i, \theta)$, where i is the control law selected and $\theta \in \Theta$ is the unknown underlying vector parametrizing the transition probabilities (θ corresponds to the transmission rate success probability vector in the original optimal link rate selection problem). For all $x, y \in \mathbb{S}$, consider $p(x, y; i, \theta)$ as follows:

$$p(x, y; i, \theta) = p(y; i, \theta) = \begin{cases} \theta_i, & \text{if } y = r_i, \\ 1 - \theta_i, & \text{if } y = 0. \end{cases}$$

Let the immediate reward $r(x, i)$ be equal to $r_i\theta_i$. Note that for any control law i, its immediate reward $r(x, i)$ is equal to its expected reward and is independent of the state x. Finding efficient adaptive sequential decision making rules in the above controlled Markov chain framework is equivalent to solving the optimal link rate selection problem. Hence, the above construction makes the optimal link rate selection problem amenable to results in [30].

Now, consider a fixed $\theta \in \Theta$. We define the set $B(\theta)$ to be the set of all bad parameters $\lambda \in \Theta$ such that when i^* is the control law chosen, λ is indistinguishable from θ, but i^* is not the optimal control law under λ:

$$B(\theta) = \{\lambda \in \Theta : \lambda_{i^*} = \theta_{i^*} \text{ and } \max_i r_i\lambda_i > r_{i^*}\lambda_{i^*}\}.$$

Consider sets $B_i(\theta), i = 1, 2, ..., n$, defined as follows:

$$B_i(\theta) = \{\lambda \in B(\theta) : r_i\lambda_i > r_{i^*}\lambda_{i^*}\}.$$

Note that $B(\theta) = \bigcup_i B_i(\theta)$. Also, note that if $r_i < r_{i^*}\theta_{i^*}$, $B_i(\theta) = \phi$. Let $P = \{i : r_i \geq r_{i^*}\theta_{i^*}\}$. Since $r_1 < r_2 < ... < r_n$, $P = \{i_1, ..., n\}$, where $i_1 \leq i^*$ is the smallest index satisfying $r_{i_1} \geq r_{i^*}\theta_{i^*}$. Define $P' = P \setminus \{i^*\}$.

Using Theorem 1 in [30], we know that $\bar{c} = (c_1, c_2, ..., c_{i^*-1}, c_{i^*+1}, ..., c_n)$, i.e., the vector of constants (in our theorem statement) for the lower bound solve the following linear program:

$$\begin{aligned} &\min \sum_i c_i(r_{i^*}\theta_{i^*} - r_i\theta_i), \\ \text{subject to } &\inf_{\lambda \in B_i(\theta)} \sum_{l \neq i^*} c_l D(\theta_l, \lambda_l) \geq 1, \forall i \in P', \\ &c_i \geq 0, \forall i, \end{aligned} \tag{2.10}$$

where $D(\theta_l, \lambda_l)$ denotes the KL-divergence between Bernoulli distributions parametrized by θ_l and λ_l. Now, all that remains to prove is that the above linear program is equivalent to the two linear programs in the theorem statement.

In order to decouple and simplify the above LP, we will focus on simplifying the first constraint. Without loss of generality, consider $i > i^*$. Note that $i \in P'$. Now, we observe the following:

1. Since $\lambda \in B_i(\theta)$, we know that $\lambda_{i^*} r_{i^*} = \theta_{i^*} r_{i^*}$ and also $\lambda_i > \{\frac{\lambda_{i^*} r_{i^*}}{r_i} = \frac{\theta_{i^*} r_{i^*}}{r_i}\} > \theta_i$. Since $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$, therefore, for any $\lambda \in B_i(\theta)$:

$$\sum_{l \neq i^*} c_l D(\theta_l, \lambda_l) \geq \sum_{l=i^*+1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}).$$

2. Consider $\lambda(\epsilon) \in B_i(\theta)$ such that $\lambda_l(\epsilon) = \theta_l, \forall l \in \{1, 2, ..., i^*\} \cup \{i+1, i+2, ..., n\}$, $\lambda_i(\epsilon) = \frac{r_{i^*}\theta_{i^*}}{r_i} + \epsilon$ and $\lambda_l(\epsilon) = \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\}\{\frac{r_{i^*}\theta_{i^*}}{r_i} + \epsilon\} + \mathbb{1}\{\theta_l > \frac{r_{i^*}\theta_{i^*}}{r_i}\}\theta_l, \forall l \in \{i^*+1, i^*+2, ..., i-1\}$. It can be easily verified that $\lambda(\epsilon) \in B_i(\theta)$. Now, using $\lambda(\epsilon)$, we get:

$$\sum_{l \neq i^*} c_l D(\theta_l, \lambda_l(\epsilon)) = \sum_{l=i^*+1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i} + \epsilon).$$

Therefore:

$$\lim_{\epsilon \to 0} \sum_{l \neq i^*} c_l D(\theta_l, \lambda_l(\epsilon)) = \sum_{l=i^*+1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}).$$

From the above facts, we can conclude that for $i > i^*$, the first constraint in the LP given by Equation (2.10) is equivalent to:

$$\sum_{l=i^*+1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}) \geq 1. \tag{2.11}$$

Similarly, for $i \in P'$ such that $i < i^*$, we can show that the first constraint in the LP given by Equation (2.10) is equivalent to:

$$\sum_{l=1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}) \geq 1. \tag{2.12}$$

Using Equations (2.11), (2.12) in the LP given by Equation (2.10), we get the following simplified LP:

$$\min \sum_i c_i (r_{i^*}\theta_{i^*} - r_i \theta_i),$$

$$\text{s. t.} \sum_{l=1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}) \geq 1, \forall i \in P', i < i^*,$$

$$\sum_{l=i^*+1}^{i} c_l \mathbb{1}\{\theta_l \leq \frac{r_{i^*}\theta_{i^*}}{r_i}\} D(\theta_l, \frac{r_{i^*}\theta_{i^*}}{r_i}) \geq 1, \forall i > i^*,$$

$$c_i \geq 0, \forall i.$$

The above LP can be very straightforwardly decoupled into two LPs as in the theorem statement, giving us the final result. □

2.4.3 Simulations

In this section, we present simulation results comparing the performance of CoTS with the current state-of-the-art MAB-based algorithms. For the optimal link rate selection problem with the basic non-increasing components structure, KL-R-UCB (see [13]) and MTS (see [15]) are the current state-of-

the-art algorithms. With the additional constraint of unimodality known, G-ORS has been shown (see [13]) to be asymptotically optimal.

We consider the same experimental setup as in [13], i.e., a single-link 802.11g system with eight available rates from 6 to 54 Mbit/s (also see [19]). The eight rates are as follows (in Mbit/s): $r_1 = 6, r_2 = 9, r_3 = 12, r_4 = 18, r_5 = 24, r_6 = 36, r_7 = 48$ and $r_8 = 54$. We implement all the algorithms in three different scenarios (different values of θ) as in [19]: *gradual*, *steep* and *lossy*. For all these scenarios, we implement and compare KL-R-UCB, MTS and CoTS (without exploiting unimodality) for the case when only the basic structure in Θ is known. We also implement and compare G-ORS and CoTS (exploiting unimodality), for the case when additional structure of unimodality is known.

Gradual

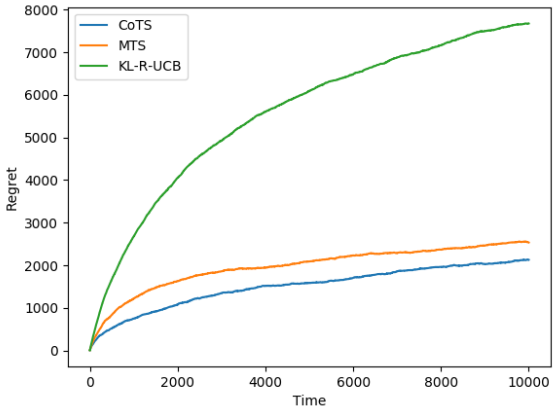
In the gradual case, we consider the following rate success probability vector: $\theta = (0.95, 0.90, 0.80, 0.65, 0.45, 0.25, 0.15, 0.10)$. Therefore, the vector of expected throughput ξ, i.e., $\xi_i = r_i\theta_i$, is: $\xi = (5.7, 8.1, 9.6, 11.7, 10.8, 9., 7.2, 5.4)$. The defining property of the gradual case is that the optimal rate is the highest rate with the probability of success greater than 0.5. Figure 2.2a compares the performance of KL-R-UCB, MTS and CoTS (without exploiting unimodality) for the gradual case. CoTS outperforms both KL-R-UCB and MTS. Another point worth noting is that both CoTS and MTS outperform KL-R-UCB by a significant margin. Figure 2.2b compares the performance of G-ORS and CoTS (exploiting unimodality) for the gradual case. Here again, CoTS outperforms G-ORS. The lower bound constant for the gradual case obtained from Theorem 2 is 526.19, whereas until $t = 10000$, CoTS achieves a constant of 154.78. Although this might seem illogical, we note that the lower bound is asymptotic. It is interesting to note that while it may take a long time to achieve the lower bound, the performance is even better than the lower bound in finite time. This is a feature we have observed in many simulations.

Steep

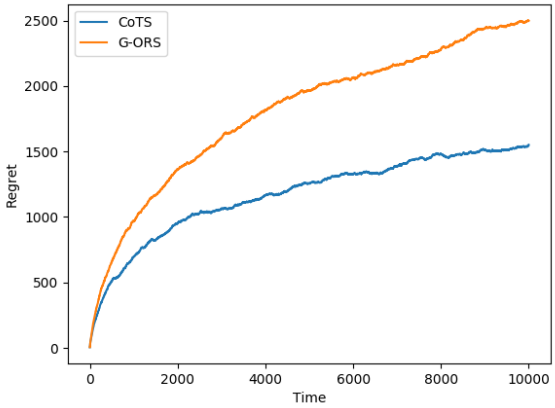
In the steep case, we consider the following rate success probability vector: $\theta = (0.99, 0.98, 0.96, 0.93, 0.90, 0.10, 0.06, 0.04)$. Therefore, the vector of expected throughput ξ is: $\xi = (5.94, 8.82, 11.52, 16.74, 21.6, 3.6, 2.88, 2.16)$. The defining property of the steep case is that the success probability of every rate is either really high or really low (either close to 1 or close to 0). Figure 2.3a compares the performance of KL-R-UCB, MTS and CoTS (without exploiting unimodality) for the steep case. Similar to the gradual case, CoTS again outperforms both KL-R-UCB and MTS. Also, CoTS and MTS again outperform KL-R-UCB by a significant margin. Figure 2.3b compares the performance of G-ORS and CoTS (exploiting unimodality) for the steep case. Here again, CoTS outperforms G-ORS. The lower bound constant for the steep case obtained from Theorem 2 is 45.56, whereas until $t = 10000$, CoTS achieves a constant of 46.49. This shows that CoTS is almost optimal.

Lossy

In the lossy case, we consider the following rate success probability vector: $\theta = (0.90, 0.80, 0.70, 0.55, 0.45, 0.35, 0.20, 0.10)$. Therefore, the vector of expected throughput ξ is: $\xi = (5.4, 7.2, 8.4, 9.9, 10.8, 12.6, 9.6, 5.4)$. The defining property of the lossy case is that the optimal rate has a low success probability, typically less than 0.5, i.e., the system loses significant packets even at the optimal rate. Figure 2.4a compares the performance of KL-R-UCB, MTS and CoTS (without exploiting unimodality) for the lossy case. Similar to the gradual and steep cases, CoTS again outperforms both KL-R-UCB and MTS. Also, CoTS and MTS again outperform KL-R-UCB by a significant margin. Figure 2.4b compares the performance of G-ORS and CoTS (exploiting unimodality) for the lossy case. Here again, CoTS outperforms G-ORS. The lower bound constant for the lossy case obtained from Theorem 2 is 401.41, whereas until $t = 10000$, CoTS achieves a constant of 181.44. Again, the performance of CoTS is better than the asymptotic lower bound in finite time.

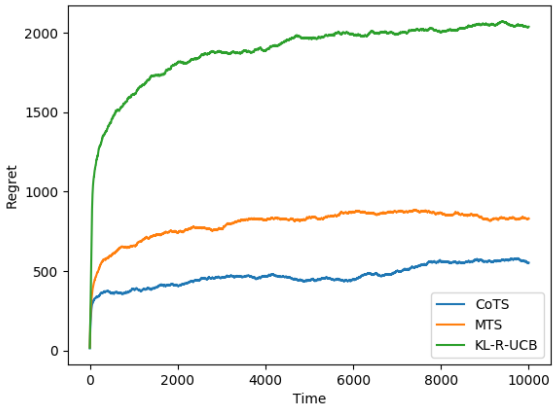


(a) Plot comparing the performance of KL-R-UCB, MTS and CoTS (*without exploiting* unimodality) for the *gradual* case.

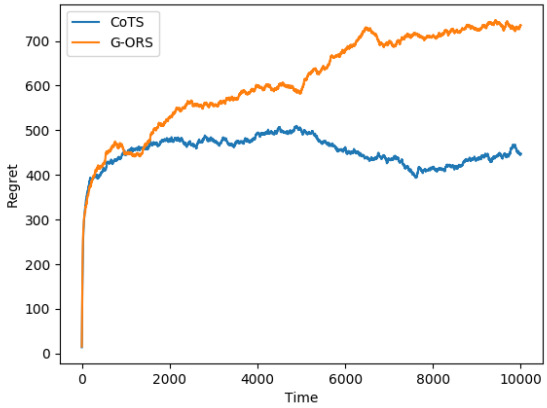


(b) Plot comparing the performance of G-ORS and CoTS (*exploiting* unimodality) for the *gradual* case.

Figure 2.2: Performance of CoTS vs. state-of-the-art in 802.11g systems with rate success probabilities characterized by the *gradual* case. Note that CoTS outperforms the current state-of-the-art in both the cases, i.e., regardless of whether one exploits unimodality or not.

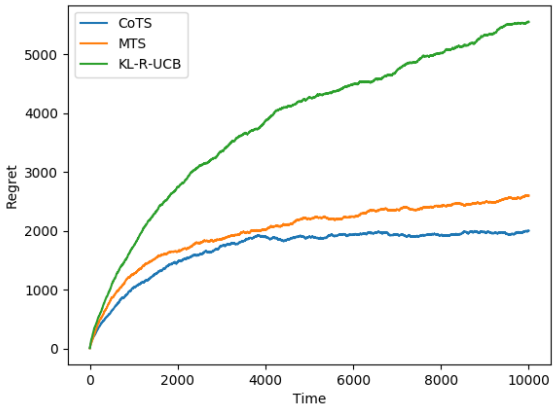


(a) Plot comparing the performance of KL-R-UCB, MTS and CoTS (*without exploiting* unimodality) for the *steep* case.

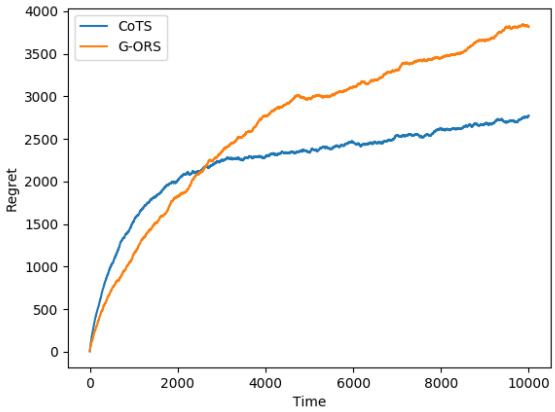


(b) Plot comparing the performance of G-ORS and CoTS (*exploiting* unimodality) for the *steep* case.

Figure 2.3: Performance of CoTS vs. state-of-the-art in 802.11g systems with rate success probabilities characterized by the *steep* case. Note that CoTS outperforms the current state-of-the-art in both the cases, i.e., regardless of whether one exploits unimodality or not.



(a) Plot comparing the performance of KL-R-UCB, MTS and CoTS (*without exploiting* unimodality) for the *lossy* case.



(b) Plot comparing the performance of G-ORS and CoTS (*exploiting* unimodality) for the *lossy* case.

Figure 2.4: Performance of CoTS vs. state-of-the-art in 802.11g systems with rate success probabilities characterized by the *lossy* case. Note that CoTS outperforms the current state-of-the-art in both the cases, i.e., regardless of whether one exploits unimodality or not.

CHAPTER 3

BANDITS WITH TWO-LEVEL FEEDBACK

3.1 Motivation

Panoramic video streaming provides an immersive experience in a virtual 3D world and has received great attention from both academia and different industries in recent years. One major challenge in high resolution panoramic video streaming is that it demands $4 \sim 6\times$ the bandwidth as compared to a regular video with the same resolution (see [31]). However, a user may just need to see roughly 20% of the entire panoramic scene without affecting her/his visual perception depending on her/his perspective. This small and relevant portion of the entire panoramic scene is known as the user's Field of View (FoV). For instance, in the case of a panoramic roller coaster video, a user can see either the front views or the back views at any given time. Therefore, if a user's motion is accurately predicted, it is sufficient to deliver just 20% of the 360° video scene surrounding him/her, thereby significantly reducing the network bandwidth consumption.

Unfortunately, it is impossible to achieve zero error in predicting a user's motion. As a result, a portion of the video larger than the FoV is usually delivered to account for the imperfect motion prediction. As long as the delivered portion covers the user's FoV, the user will be able to successfully view the content. Although a larger delivery portion can tolerate a larger prediction error and thus yield a higher probability of prediction (FoV coverage), it can result in transmission failure due to the fact that a larger portion of the video may exceed the maximum transmission rate at the current channel state. Thus, a central question is how to select an appropriate delivery portion at each time to maximize system throughput or some other metric of the user's quality of experience.

Recent works (e.g., [31, 32, 33, 34, 35, 36, 37]) have explored various ef-

ficient user motion prediction algorithms and have used them to aid adaptive delivery portion selection for panoramic video streaming. These papers typically require the collection of head motion traces from many users for different video content, and subsequently train a motion prediction model on the collected data. However, they neither explore fast-changing wireless environments nor adapt to user's personal behavior and panoramic video content. To this end, in this thesis, we formulate the delivery portion selection problem as a stochastic multi-armed bandit (MAB) problem, where different delivery portions of the panoramic scene correspond to different arms and the goal is to minimize the regret (i.e., the gap between the optimal cumulative throughput and the cumulative throughput under an algorithm) over a finite time horizon. The considered setup has two-level feedback, i.e., after each arm is played, we receive both the prediction and transmission outcomes and the reward is determined by the product of these two independent pieces of information.

The MAB problem is a well-studied problem and has a wide array of practical applications (see [38] for reference). In the traditional stochastic MAB setting, at each time slot, an agent plays an arm (from a set of arms) and receives a random reward drawn (independently across time) from the reward distribution of the arm it played. The goal of the agent is to minimize its regret over a certain time horizon (defined as the loss in expected reward incurred by the agent as compared to an oracle which knows the optimal arm) by taking sequential decisions which strike a delicate balance between exploiting the information that the agent already has and exploring different arms in order to gain more information. The fundamental difference between our problem and the standard stochastic MAB problem is that our reward is determined by the product of two independent pieces of random information instead of simply one net reward feedback. Although we can consider the product of the two levels of feedback as the net reward and formulate the problem as the standard stochastic MAB problem, we wish to exploit the potentially higher level of information that the two independent levels of feedback can provide as compared to their product.

In their seminal work [39], the authors proved a fundamental lower bound on the regret that can be achieved by any uniformly good algorithm for the traditional stochastic MAB problem. Since then, a number of popular and easy to implement algorithms have been designed which asymptotically

achieve this fundamental lower bound (e.g., KL-UCB [26] and Thompson sampling [40]). While some recent works (e.g., [41, 42]) have considered the bandit problem with multiple-level feedback, they did not explore the gain of information between multiple-level feedback and one-level feedback. Our main contributions in this thesis are as follows:

1. We formulate the problem of maximizing the system throughput in panoramic video delivery as a stochastic multi-armed bandit (MAB) problem with two-level feedback. This non-standard formulation of the stochastic MAB problem can be more generally useful in other MAB application domains where multiple levels of feedback are available (see Section 3.2).

2. We present and prove a lower bound on the regret of any uniformly good algorithm (formally defined in Section 3.3) for the stochastic MAB problem with two-level feedback. We show that this lower bound is always at least as small as the lower bound for the standard MAB problem with single feedback, illustrating the potential advantage of using the multiple levels of feedback independently (see Section 3.3).

3. Using insights obtained from the lower bound, we design a KL-UCB-style algorithm to efficiently solve the stochastic MAB problem with two-level feedback. We theoretically analyze the performance of our algorithm and show that it is asymptotically optimal (see Section 3.4).

4. In order to establish the practical utility of our algorithm, we conduct experiments on synthetic data as well as data obtained from real traces. We conclusively establish that our algorithm consistently outperforms the standard KL-UCB algorithm (which is known to be optimal for the single feedback problem, see Section 3.5).

3.2 Problem Formulation

We consider a single user downloading a 360° video from an access point (AP) over a wireless channel, as shown in Fig. 3.1. We assume that the system operates in slotted time with normalized time slots $j \in \{1, 2, \ldots, n\}$. In each time slot j, only a portion (typically 20%) of the whole panoramic scene

can be seen by a user, namely the *Field of View (FoV)*. If we can accurately predict a user's head movement, then it is sufficient to deliver just 20% of the whole scene, which consumes only 1/5th of the originally required wireless bandwidth. Unfortunately, a user typically randomly moves his/her head depending on his/her interest in the video content. Hence, it is unavoidable to incur errors in head motion prediction. To mitigate this, we usually deliver a portion of the video larger than the FoV.

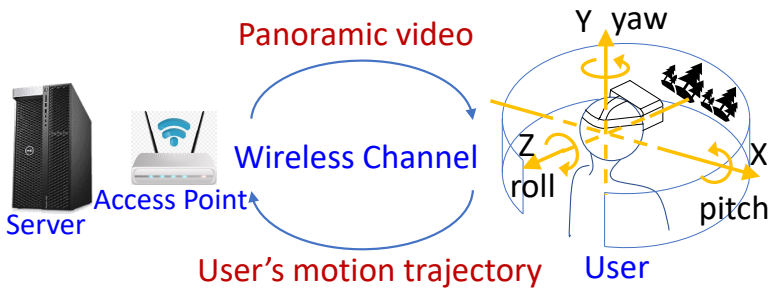


Figure 3.1: Wireless panoramic video streaming system

We note that each panoramic video is usually partitioned into a series of chunks, where each chunk has a fixed number of tiles of the same duration. In each time slot, a subset of tiles can be selected for transmission. Since there are a finite number of subsets of tiles in each video chunk, we assume that there are K different rates corresponding to different portions of the panoramic scenes with $0 < r_1 < r_2 < \cdots < r_K$, where r_1 refers to the rate when only the predicted FoV is delivered and r_K corresponds to the rate that delivers the whole panoramic video. We use $X_i(j) = 1$ to denote that user's FoV is covered by the delivered portion in time slot j when rate r_i is used ($X_i(j) = 0$ otherwise). Let $\alpha_i \triangleq \Pr\{X_i(j) = 1\}$ be the *prediction probability*. Note that the AP gets to know the user's exact FoV after each transmission, even if the transmission fails, since the user's device automatically records the user's current motion orientation (yaw, pitch, roll) and sends that information back to the AP. Hence, if rate r_i was used in time slot j for transmission, the AP always knows the outcome of $X_i(j)$.

We assume that user's channel rate is independently and identically distributed (i.i.d.) over time. We assume that channel rate is unknown at the beginning of each time slot. If the selected rate is less than or equal to the channel rate, then the wireless transmission will succeed. Otherwise, the transmission will fail. We use $Y_i(j) = 1$ to denote a successful transmission

when the rate r_i is used in time slot j ($Y_i(j) = 0$ if the transmission fails). Let $\beta_i \triangleq \Pr\{Y_i(j) = 1\}$ be the *transmission probability*.

Let $i(t) \in \{1, 2, ..., K\}$ denote the index of the rate used for transmission at time slot t. Note that the user can successfully view the desired content only when both the prediction and the transmission are successful, i.e., $X_{i(t)}(t)Y_{i(t)}(t) = 1$. Hence, the user's throughput in time slot t is $r_{i(t)}X_{i(t)}(t)Y_{i(t)}(t)$. Our model can also accommodate other panoramic video coding schemes, where the portion containing the FoV is encoded with a high resolution and the rest of scenes are encoded with lower resolution. In such a regime, rate r_i can be viewed as a measure of the user's QoE (quality of experience) and a transmission is considered to fail if the intended QoE is not delivered to the user.

In this thesis, the AP needs to make a decision on the selected rate in order to maximize the system throughput (or some other metric of QoE as mentioned above). If both the user's prediction and transmission probabilities (i.e., $\{\alpha_i, \beta_i, i = 1, 2, \ldots, K\}$) are known, then the optimal throughput can be achieved by solving the following optimization problem: $i^* \in \arg\max_{i=1,2,\ldots,K} r_i\alpha_i\beta_i$. However, both the prediction and transmission probabilities are unknown, since they rely on many factors such as the wireless environment, the panoramic video content, and the user's personal behavior. Thus, the algorithm not only needs to learn these statistics (also known as (a.k.a.) exploration) but also to select the best rate so far (a.k.a. exploitation). Our goal is to design a learning algorithm that achieves the maximum system throughput within n time slots, where n is some positive integer. This is equivalent to minimizing the regret, which is the gap between the expected accumulated throughput and the optimal throughput, i.e.,

$$\begin{aligned} R(n) &\triangleq nr_{i^*}\alpha_{i^*}\beta_{i^*} - \mathbb{E}\left[\sum_{j=1}^{n} r_{i(j)}X_{i(j)}(j)Y_{i(j)}(j)\right] \\ &= \sum_{k\neq i^*} \mathbb{E}\left[T_k(n)\Delta_k\right], \end{aligned} \tag{3.1}$$

where $\Delta_k = r_{i^*}\alpha_{i^*}\beta_{i*} - r_k\alpha_k\beta_k$ and $T_k(n)$ denotes the number of times rate r_k was used as the rate of transmission until the end of time slot n.

Different from the traditional multi-armed bandit problem (where only the product $X_{i(t)}(t)Y_{i(t)}(t)$ is available as feedback), both prediction and trans-

mission outcomes (i.e., $X_{i(t)}(t)$ and $Y_{i(t)}(t)$) are available to us after each decision on the selected rate. This additional level of feedback information can indeed be efficiently leveraged to reduce the regret compared with the single feedback counterpart. To that end, we first characterize the fundamental lower bound on the regret if two-level feedback information is available and compare it with that in the traditional multi-armed bandit problem with single feedback information. Note that, in this thesis, we do not consider any possible structure in $\{\alpha_i, i = 1, 2, ..., K\}$ and $\{\beta_i, i = 1, 2, ..., K\}$ which can further possibly improve throughput, in addition to exploiting two-level feedback. We leave the task of incorporating such structure to future work (see [10], [43] and references therein for literature on structured bandit problems).

3.3 Lower Bound on the Regret

In this section, we derive a lower bound on the regret (as defined in Equation (3.1)) achieved by any uniformly good algorithm for the stochastic multi-armed bandit problem with two-level feedback. We use the analysis technique from [39] and adapt it to our problem setup, similar to [44]. We adopt notation similar to [44] for ease of exposition.

Theorem 3. *For the stochastic multi-armed bandit problem with two-level feedback, let ψ be any uniformly good algorithm, i.e., as $n \to \infty$, the regret achieved by ψ (for any valid problem instance) belongs to the set $o(n^\delta)$,[1] for any $\delta \in (0, 1)$. Then, the following result holds for the regret achieved by ψ:*

$$\liminf_{n\to\infty} \frac{R(n)}{\log n} \geq \sum_{k\neq i^*} \frac{\Delta_k \mathbb{1}\{\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k} < 1\}}{\min\limits_{\substack{0\leq x,y\leq 1 \\ xy\geq \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k}}} d(\alpha_k, x) + d(\beta_k, y)},$$

where $d(a, b) \triangleq a \log \frac{a}{b} + (1-a) \log \frac{1-a}{1-b}$ denotes the KL-divergence between two Bernoulli random variables with mean a and b respectively, and $\mathbb{1}\{\mathcal{A}\} = 1$ if event $\mathcal{A}$ is true and 0 otherwise.

[1] $f(n) \in o(g(n))$ if for all $c > 0$, there exists some $k > 0$ such that $0 \leq f(n) < cg(n), \forall n \geq k$

Proof. Let the original vector of prediction and transmission success probabilities be denoted by $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta} \triangleq (\theta_1, \theta_2, ..., \theta_K)$, where $\theta_i = (\alpha_i, \beta_i)$. Note that we assume a unique optimal rate, i.e., $r_{i^*}\alpha_{i^*}\beta_{i^*} > r_i\alpha_i\beta_i, \forall i \neq i^*$. Next, we consider a sub-optimal rate r_k with parameter $\theta_k = (\alpha_k, \beta_k)$, i.e, $k \neq i^*$, and lower bound the number of times transmissions at rate r_k (i.e., $T_k(n)$) are made by any uniformly good algorithm ψ (as defined in the theorem). Using standard change of measure arguments as in [39], we can show that if $r_{i^*}\alpha_{i^*}\beta_{i^*} < r_k$, then:

$$\lim_{n\to\infty} \mathbb{P}_{\boldsymbol{\theta}}\left(T_k(n) \leq \frac{(1-\chi)(1-\delta)\log n}{(1+\gamma)\big(d(\alpha_k, x) + d(\beta_k, y)\big)}\right) = 0, \tag{3.2}$$

where $0 < \delta, \chi < 1$, $\gamma > 0$, and $0 \leq x, y \leq 1$ satisfies $r_k xy > r_{i^*}\alpha_{i^*}\beta_{i^*}$. If $r_{i^*}\alpha_{i^*}\beta_{i^*} \geq r_k$, then $\lim_{n\to\infty} \mathbb{P}_{\boldsymbol{\theta}}\big(T_k(n) \leq c \times o(\log n)\big) = 0$, where c is a problem-dependent constant. Note that only the case $r_{i^*}\alpha_{i^*}\beta_{i^*} < r_k$ is non-trivial. To get the tightest lower bound, we optimize the upper bound inside Equation (3.2) to get:

$$\begin{aligned}\lim_{n\to\infty} \mathbb{P}_{\boldsymbol{\theta}}\left(T_k(n) \leq \frac{(1-\chi)(1-\delta)\log n}{(1+\gamma)F^{(\upsilon)}(\alpha_k, \beta_k)}\right) \\ \leq \lim_{n\to\infty} \mathbb{P}_{\boldsymbol{\theta}}\left(T_k(n) \leq \frac{(1-\chi)(1-\delta)\log n}{(1+\gamma)F(\alpha_k, \beta_k)}\right) = 0,\end{aligned}$$

where

$$\begin{aligned}F(\alpha_k, \beta_k) &= \min_{\substack{0\leq x,y\leq 1 \\ xy > r_{i^*}\alpha_{i^*}\beta_{i^*}/r_k}} d(\alpha_k, x) + d(\beta_k, y), \\ F^{(\upsilon)}(\alpha_k, \beta_k) &= \min_{\substack{0\leq x,y\leq 1, \upsilon > 0 \\ xy \geq r_{i^*}\alpha_{i^*}\beta_{i^*}/r_k + \upsilon}} d(\alpha_k, x) + d(\beta_k, y),\end{aligned}$$

and we use the fact that $F^{(\upsilon)}(\alpha_k, \beta_k) \geq F(\alpha_k, \beta_k)$. This implies

$$\lim_{n\to\infty} \mathbb{P}_{\boldsymbol{\theta}}\left(T_k(n) > \frac{(1-\chi)(1-\delta)\log n}{(1+\gamma)F^{(\upsilon)}(\alpha_k, \beta_k)}\right) = 1. \tag{3.3}$$

Since δ, χ, γ and υ are positive variables that can be chosen to be arbitrarily close to 0, we can use results 2 and 3 from Lemma 1 (presented after this theorem) along with simple algebraic manipulations to write the above

equation succinctly as follows:

$$\lim_{n\to\infty} \mathbb{P}_{\boldsymbol{\theta}}\left(T_k(n) \geq \frac{(1-\epsilon)\log n}{\min\limits_{\substack{0\leq x,y\leq 1 \\ xy\geq \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k}}} d(\alpha_k, x) + d(\beta_k, y)}\right) = 1,$$

for a sufficiently small $\epsilon > 0$. The desired result follows straightforwardly from the above equation (see [39] for more details). $\square$

We now present and prove an important lemma that quantifies the effect that small perturbations in the constraints of the optimization problem in the denominator of the lower bound have on its solution. In addition to being useful in the proof of the lower bound, this lemma will also be extremely useful in our analysis for establishing an upper bound on the regret achieved by the KL-UCB-style algorithm that we design (Section 3.4).

Lemma 1. *For $0 \leq x, y < 1$ and a constant c such that $xy < c \leq 1$, consider the following optimization problems:*

$$\big(p^*(x,y), q^*(x,y)\big) = \arg \min_{0\leq p,q\leq 1;\ pq\geq c} d(x,p) + d(y,q),$$
$$\big(p^*_{-\epsilon}(x,y), q^*_{-\epsilon}(x,y)\big) = \arg \min_{0\leq p,q\leq 1;\ pq\geq c-\epsilon} d(x,p) + d(y,q).$$

For $0 \leq x, y < 1$ and a constant c such that $xy < c < 1$, consider the following optimization problem:

$$\big(p^*_{+\epsilon}(x,y), q^*_{+\epsilon}(x,y)\big) = \arg \min_{0\leq p,q\leq 1;\ pq\geq c+\epsilon} d(x,p) + d(y,q).$$

The following results hold for the solutions to the above three optimization problems:

1. *$p^*(x,y) > x$ and $q^*(x,y) > y$.*
2. *Let $h_c(x,y) \triangleq \sqrt{(x-y)^2 + 4c(1-x)(1-y)}$. The following results hold:*

(a) *For* $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, \frac{c-xy}{1+x+y}\}$*:*

$$\min_{\substack{\alpha,\beta\in[x-\epsilon,x+\epsilon]\\ \times[y-\epsilon,y+\epsilon]}} p^*(\alpha,\beta) \geq p^*(x,y) - \frac{1+\frac{4}{h_c(x,y)}}{1-y}\epsilon,$$

$$\min_{\substack{\alpha,\beta\in[x-\epsilon,x+\epsilon]\\ \times[y-\epsilon,y+\epsilon]}} q^*(\alpha,\beta) \geq q^*(x,y) - \frac{1+\frac{4}{h_c(x,y)}}{1-x}\epsilon.$$

(b) *For* $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, c-xy\}$*:*

$$p^*_{-\epsilon}(x,y) \geq p^*(x,y) - \frac{2(1-x)}{h_c(x,y)}\epsilon,$$
$$\text{and } q^*_{-\epsilon}(x,y) \geq q^*(x,y) - \frac{2(1-y)}{h_c(x,y)}\epsilon.$$

(c) *For* $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, 1-c\}$*:*

$$p^*_{+\epsilon}(x,y) \geq p^*(x,y) + \frac{1-x}{2h_c(x,y)}\epsilon,$$
$$\text{and } q^*_{+\epsilon}(x,y) \geq q^*(x,y) + \frac{1-y}{2h_c(x,y)}\epsilon.$$

Additionally, the following bounds quantify the impact of perturbations on the KL-divergence between two Bernoulli random variables.

3. *Let* $\beta > \alpha$*.*

(a) *For* $\epsilon_1 \in [0, \frac{1-\alpha}{2}]$ *and* $\epsilon_2 \in [0, 1-\beta]$ *such that* $\epsilon_1 + \epsilon_2 < \beta - \alpha$*, the following result holds:*

$$d(\alpha+\epsilon_1, \beta-\epsilon_2) \geq d(\alpha,\beta) - c_1\epsilon_1 - c_2\epsilon_2,$$

where $c_1 = \log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} + 2$ *and* $c_2 = \frac{1-\alpha}{1-\beta}$*.*

(b) *For* $\epsilon_1 \in [0, \min\{\frac{\alpha}{2}, \frac{1-\alpha}{2}\}]$ *and* $\epsilon_2 \in [0, \min\{\frac{\beta}{2}, \frac{1-\beta}{2}\}]$*, the following result holds:*

$$d(\alpha-\epsilon_1, \beta+\epsilon_2) \leq d(\alpha,\beta) + c_1'\epsilon_1 + c_2'\epsilon_2,$$

where $c_1' = \log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} + 4$ *and* $c_2' = \frac{2(1-\alpha)}{(1-\beta)}$*.*

Proof. Although the constraint on $p^*(x,y)$ and $q^*(x,y)$ is that $p^*(x,y)\times q^*(x,y) \geq c$, we will show that $p^*(x,y)q^*(x,y) = c$. To this end, let us assume to the contrary that $p^*(x,y)q^*(x,y) = c' > c$. Since $c' > c > xy$, one of the following is true: $p^*(x,y) > x$ or $q^*(x,y) > y$. Without loss of generality, let us assume that $p^*(x,y) > x$. Since $p^*(x,y)q^*(x,y) = c' > c$, $\exists \xi > 0$ such that $c < (p^*(x,y) - \xi)q^*(x,y) < c'$ and $p^*(x,y) - \xi > x$. Therefore, $d(p^*(x,y) - \xi, x) + d(q^*(x,y), y) < d(p^*(x,y), x) + d(q^*(x,y), y)$. Clearly, this is a contradiction. Therefore, $p^*(x,y)q^*(x,y) = c$ and thus we can rewrite the optimization problem as:

$$\begin{aligned} p^*(x,y) =& \arg\min_{c\leq p\leq 1} d(x,p) + d(y,\frac{c}{p}), \\ q^*(x,y) =& \frac{c}{p^*(x,y)}. \end{aligned}$$

Let $l(p) = d(x,p) + d(y,\frac{c}{p})$. Therefore:

$$\begin{aligned} l(p) =& x\log\frac{x}{p} + (1-x)\log\frac{(1-x)}{(1-p)} + y\log\frac{yp}{c} + (1-y)\log\frac{(1-y)p}{p-c} \\ \Rightarrow l'(p) =& -\frac{x}{p} + \frac{1-x}{1-p} + \frac{y}{p} + \frac{(y-1)c}{(p-c)p}. \end{aligned}$$

Observe that at the boundary points in the domain of p (i.e., $p = c$ and $p = 1$), $l(p) = \infty$. Therefore, if $c < 1$, we simply set $l'(p^*(x,y)) = 0$ to obtain:

$$p^*(x,y) = \frac{x - y + \sqrt{(x-y)^2 + 4c(1-y)(1-x)}}{2(1-y)}. \tag{3.4}$$

For $c = 1$, $p^*(x,y) = 1$ trivially. Conveniently, the above equation also holds for $c = 1$. Since $c > xy$, and $0 \leq x, y < 1$, we have:

$$\begin{aligned} p^*(x,y) &> \frac{x - y + \sqrt{(x-y)^2 + 4xy(1-y)(1-x)}}{2(1-y)} \\ &= \frac{x - y + \sqrt{(x+y-2xy)^2}}{2(1-y)} \\ &= \frac{x - y + x + y - 2xy}{2(1-y)} = x, \end{aligned}$$

where the penultimate equality follows from the fact that $x + y \geq 2\sqrt{xy} \geq$

$2xy, \forall\ 0 \leq x, y \leq 1$. A similar analysis can be done to show $q^*(x,y) > y$ by simply swapping x and y in the above analysis. This proves the first part of the lemma.

Next, we prove that for $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, \frac{c-xy}{1+x+y}\}$:

$$\min_{\alpha,\beta\in[x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]} p^*(\alpha,\beta) \geq p^*(x,y) - \frac{1+\frac{4}{h_c(x,y)}}{1-y}\epsilon.$$

Note that the condition on ϵ implies that $c > (x+\epsilon)(y+\epsilon)$, therefore, $p^*(\alpha,\beta), \forall \alpha,\beta \in [x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]$ can be solved using Equation (3.4). To prove the above inequality, we first show that $\min_{\alpha,\beta\in[x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]} p^*(\alpha,\beta)$ $= p^*(x-\epsilon, y+\epsilon)$, by proving that $\frac{dp^*(\alpha,\beta)}{d\alpha} \geq 0$ and $\frac{dp^*(\alpha,\beta)}{d\beta} \leq 0, \forall \alpha,\beta \in [x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]$. Using Equation (3.4):

$$\frac{dp^*(\alpha,\beta)}{d\alpha} = \frac{h_c(\alpha,\beta)+\alpha-\beta-2c(1-\beta)}{2(1-\beta)h_c(\alpha,\beta)}, \tag{3.5}$$

where $h_c(\alpha,\beta) = \sqrt{(\alpha-\beta)^2+4c(1-\alpha)(1-\beta)}$. We want to show that the expression on the RHS is always greater than or equal to zero. The denominator in the above equation is always positive, therefore, we need to simply show that the numerator is always greater than or equal to zero. For any $\alpha,\beta \in [x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]$, let the numerator be denoted by: $g_1(c) = h_c(\alpha,\beta)+\alpha-\beta-2c(1-\beta)$. Observe that $g_1(c)$ is a concave function in c. Also, note that $g_1(\alpha\beta) = 2\alpha(\beta-1)^2 \geq 0$ and $g_1(1) = 0$. For any $\alpha\beta < c \leq 1$, $\exists\lambda \in [0,1]$ such that $c = \lambda\alpha\beta + (1-\lambda)$. Therefore, for any $\alpha\beta < c \leq 1$, $g_1(c) \geq \lambda g_1(\alpha\beta) + (1-\lambda)g_1(1) \geq 0$. Therefore, $\frac{dp^*(\alpha,\beta)}{d\alpha} \geq 0$. Now, we need to show that $\frac{dp^*(\alpha,\beta)}{d\beta} \leq 0$. Using Equation (3.4):

$$\frac{dp^*(\alpha,\beta)}{d\beta} = \frac{(\alpha-1)h_c(\alpha,\beta)+(\alpha-\beta)(\alpha-1)+2c(1-\alpha)(1-\beta)}{2(1-\beta)^2h_c(\alpha,\beta)}. \tag{3.6}$$

In order to show $\frac{dp^*(\alpha,\beta)}{d\beta} \leq 0$, we proceed similarly as we did to show that $\frac{dp^*(\alpha,\beta)}{d\alpha} \geq 0$ and prove that $\frac{-dp^*(\alpha,\beta)}{d\beta} \geq 0$. The denominator of $\frac{-dp^*(\alpha,\beta)}{d\beta}$ is always positive, therefore, we need to simply show that the numerator is always greater than or equal to zero. For any $\alpha,\beta \in [x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]$, let the numerator be denoted by $g_2(c) = -\big\{(\alpha-1)h_c(\alpha,\beta)+(\alpha-\beta)(\alpha-1)+2c(1-\alpha)(1-\beta)\big\}$. Observe that $g_2(c)$ is a concave function in c. Also, note

that $g_2(\alpha\beta) = 2\alpha(1-\alpha)(\beta-1)^2 \geq 0$ and $g_2(1) = 0$. For any $\alpha\beta < c \leq 1$, $\exists \lambda \in [0,1]$ such that $c = \lambda\alpha\beta + (1-\lambda)$. Therefore, for any $\alpha\beta < c \leq 1$, $g_2(c) \geq \lambda g_2(\alpha\beta) + (1-\lambda) g_2(1) \geq 0$. Therefore, $\frac{-dp^*(\alpha,\beta)}{d\beta} \geq 0$ which implies $\frac{dp^*(\alpha,\beta)}{d\beta} \leq 0$. Hence, we have: $\min_{\alpha,\beta\in[x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]} p^*(\alpha,\beta) = p^*(x-\epsilon, y+\epsilon)$. Therefore, using Equation (3.4):

$$
\begin{aligned}
&\min_{\substack{\alpha,\beta\in[x-\epsilon,x+\epsilon]\\ \times[y-\epsilon,y+\epsilon]}} p^*(\alpha,\beta) \\
&\qquad = p^*(x-\epsilon, y+\epsilon) \\
&\qquad = \frac{x-y-2\epsilon+\sqrt{(x-y-2\epsilon)^2 + 4c(1-x+\epsilon)(1-y-\epsilon)}}{2(1-y-\epsilon)} \\
&\qquad \overset{(a)}{\geq} \frac{x-y-2\epsilon+\sqrt{h_c^2(x,y) + 4\epsilon^2(1-c) + 4\epsilon(x-y)(c-1)}}{2(1-y)} \\
&\qquad \geq \frac{x-y-2\epsilon+\sqrt{h_c^2(x,y) + 4\epsilon(x-y)(c-1)}}{2(1-y)} \\
&\qquad \overset{(b)}{\geq} \frac{x-y-2\epsilon+\sqrt{h_c^2(x,y) - 8\epsilon}}{2(1-y)} \\
&\qquad \overset{(c)}{\geq} \frac{x-y-2\epsilon+h_c(x,y)\left\{1-\frac{8\epsilon}{h_c^2(x,y)}\right\}}{2(1-y)} \\
&\qquad \geq \frac{x-y+h_c(x,y)}{2(1-y)} - \frac{2\epsilon+\frac{8\epsilon}{h_c(x,y)}}{2(1-y)} \\
&\qquad = p^*(x,y) - \frac{1+\frac{4}{h_c(x,y)}}{1-y}\epsilon,
\end{aligned}
$$

where step (a) follows from the definition $h_c(x,y)$, i.e., $h_c^2(x,y) = (x-y)^2 + 4c(1-x)(1-y)$; (b) follows from the fact that $(x-y)(c-1) \geq -2$; (c) follows from the inequality $\sqrt{1-x} \geq 1-x, \forall x \in [0,1]$ by choosing ϵ such that $\frac{8\epsilon}{h_c^2(x,y)} < 1$. A similar analysis can be done to show that for $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, \frac{c-xy}{1+x+y}\}$:

$$
\min_{\alpha,\beta\in[x-\epsilon,x+\epsilon]\times[y-\epsilon,y+\epsilon]} q^*(\alpha,\beta) \geq q^*(x,y) - \frac{1+\frac{4}{h_c(x,y)}}{1-x}\epsilon.
$$

Now, we prove that for $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, c-xy\}$, $p^*_{-\epsilon}(x,y) \geq p^*(x,y) - \frac{2(1-x)}{h_c(x,y)}\epsilon$. Note that the conditions on ϵ imply that we can use Equation (3.4) to solve

for $p^*_{-\epsilon}(x,y)$. Therefore:

$$\begin{aligned}
p^*_{-\epsilon}(x,y) =& \frac{x-y+\sqrt{(x-y)^2+4(c-\epsilon)(1-y)(1-x)}}{2(1-y)} \\
=& \frac{x-y+h_c(x,y)\sqrt{1-\frac{4\epsilon(1-x)(1-y)}{h_c^2(x,y)}}}{2(1-y)} \\
\overset{(a)}{\geq}& \frac{x-y+h_c(x,y)\big\{1-\frac{4\epsilon(1-x)(1-y)}{h_c^2(x,y)}\big\}}{2(1-y)} \\
=& p^*(x,y) - \frac{2(1-x)}{h_c(x,y)}\epsilon,
\end{aligned}$$

where the penultimate inequality (a) follows from the inequality $\sqrt{1-x} \geq 1-x, \forall x \in [0,1]$ and the fact that $\frac{4\epsilon(1-x)(1-y)}{h_c^2(x,y)} < \frac{8\epsilon}{h_c^2(x,y)} < 1$. A similar analysis can be done to prove that for $\epsilon < \min\{\frac{h_c^2(x,y)}{8}, c-xy\}$: $q^*_{-\epsilon}(x,y) \geq q^*(x,y) - \frac{2(1-y)}{h_c(x,y)}\epsilon$. We can prove part 2(c) of the lemma in a similar fashion as above.

Next, we prove that for $0 \leq \epsilon_1 \leq \frac{1-\alpha}{2}$ and $0 \leq \epsilon_2 \leq 1-\beta$ such that $\epsilon_1+\epsilon_2 < \beta-\alpha$, if $\beta > \alpha$, then $d(\alpha+\epsilon_1, \beta-\epsilon_2) \geq d(\alpha,\beta) - c_1\epsilon_1 - c_2\epsilon_2$, where $c_1 = \log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} + 2$ and $c_2 = \frac{1-\alpha}{1-\beta}$. Consider:

$$\begin{aligned}
&d(\alpha+\epsilon_1, \beta-\epsilon_2) \\
&\quad= (\alpha+\epsilon_1)\log\frac{\alpha+\epsilon_1}{\beta-\epsilon_2} + (1-\alpha-\epsilon_1)\log\frac{1-\alpha-\epsilon_1}{1-\beta+\epsilon_2} \\
&\quad= d(\alpha,\beta) + \alpha\log\frac{1+\frac{\epsilon_1}{\alpha}}{1-\frac{\epsilon_2}{\beta}} + (1-\alpha)\log\frac{1-\frac{\epsilon_1}{1-\alpha}}{1+\frac{\epsilon_2}{1-\beta}} \\
&\qquad+ \epsilon_1\log\big(\frac{(1+\frac{\epsilon_1}{\alpha})(1+\frac{\epsilon_2}{1-\beta})}{(1-\frac{\epsilon_2}{\beta})(1-\frac{\epsilon_1}{1-\alpha})}\big) - \epsilon_1\log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} \\
&\quad\geq d(\alpha,\beta) - \epsilon_1\log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} + (1-\alpha)\log(1-\frac{\epsilon_1}{1-\alpha}) \\
&\qquad- (1-\alpha)\log(1+\frac{\epsilon_2}{1-\beta}) \\
&\quad\geq d(\alpha,\beta) - \epsilon_1\log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} - \frac{\epsilon_1(1-\alpha)}{1-\alpha-\epsilon_1} - \frac{(1-\alpha)\epsilon_2}{1-\beta} \\
&\quad\geq d(\alpha,\beta) - \epsilon_1\big(\log\frac{\beta(1-\alpha)}{\alpha(1-\beta)} + 2\big) - \epsilon_2\frac{1-\alpha}{1-\beta},
\end{aligned}$$

where the penultimate inequality as well as the last inequality follow from

the constraints on ϵ_1, ϵ_2 combined with the following fact: $\frac{x}{1+x} \leq \log(1+x) \leq x, \forall x > -1$. A similar analysis can be done to prove part 3(b) of the lemma.

This concludes the proof of the lemma. □

We now present a corollary to Theorem 3 that shows that the lower bound on the regret with two-level feedback information is less than or equal to the standard lower bound on the regret with single feedback information (i.e., treating the product as the feedback).

Corollary 1.

$$\sum_{k \neq i^*} \frac{\Delta_k \mathbb{1}\{\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k} < 1\}}{\min\limits_{\substack{0 \leq x,y \leq 1 \\ xy \geq \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k}}} d(\alpha_k, x) + d(\beta_k, y)}$$
$$\leq \sum_{k \neq i^*} \frac{\Delta_k \mathbb{1}\{\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k} < 1\}}{\min\limits_{\substack{0 \leq x,y \leq 1 \\ xy \geq \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k}}} d(\alpha_k\beta_k, xy)}. \tag{3.7}$$

Proof. It suffices to show that

$$d(\alpha_k, x) + d(\beta_k, y) \geq d(\alpha_k\beta_k, xy). \tag{3.8}$$

We consider four independent Bernoulli random variables $X_1 \sim Ber(\alpha_k)$, $Y_1 \sim Ber(\beta_k)$, $X_2 \sim Ber(x)$, and $Y_2 \sim Ber(y)$. By considering the two random vectors $Z_1 = (X_1, Y_1)$ and $Z_2 = (X_2, Y_2)$, from the additive property of the KL-divergence (see [45] for details) for independent random variables, we get

$$\begin{aligned} KL(Z_1||Z_2) &= KL(X_1||X_2) + KL(Y_1||Y_2) \\ &= d(\alpha_k, x) + d(\beta_k, y), \end{aligned} \tag{3.9}$$

where we have used the notation KL to denote the KL-divergence between the distributions of random vectors or random variables and $d(a, b)$ is the KL-divergence between two Bernoulli random variables with means a and b. Next, consider a channel which takes two Bernoulli random variables X, Y as input and produces the output XY. Suppose we give $Z_1 = (X_1, Y_1)$ and $Z_2 = (X_2, Y_2)$ as inputs to this channel, the KL-divergence between the outputs will be less than or equal to the KL-divergence between the corresponding

inputs, a property known as the data processing inequality in information theory [45]. Thus, we have

$$KL(Z_1||Z_2) \geq KL(X_1Y_1||X_2Y_2) = d(\alpha_k\beta_k, xy). \tag{3.10}$$

By combining Equations (3.9) and (3.10), we have the desired result.

□

Remark 2. *We would like to point out that depending on the values of* $\{\alpha_i, \beta_i, r_i; i = 1, 2, ..., K\}$*, the difference between the lower bound in the two-level feedback case and the lower bound in the traditional single feedback case can be significant. Therefore, there is substantial potential in exploiting the multiple levels of feedback available to us.*

3.4 Algorithm Design and Analysis

In this section, we will propose an algorithm that achieves the lower bound on the regret derived in Section 3.3. Our proposed algorithm extends the well-known Kullback-Leibler Upper Confidence Bound (KL-UCB) algorithm (see [38, Chapter 10]) and efficiently leverages two-level feedback information.

To describe our algorithm, we will define certain quantities. Recall that $i(t)$ denotes the index of the rate selected for transmission at time slot t and $T_i(t)$ denotes the number of times that rate r_i is selected, until time slot t. Let $S_{i,1}(t) \triangleq \sum_{j=1}^{t} X_{i(t)}(t)\mathbb{1}\{i(t) = i\}$ and $S_{i,2}(t) \triangleq \sum_{j=1}^{t} Y_{i(t)}(t)\mathbb{1}\{i(t) = i\}$ respectively denote the number of times that the prediction is successful and the number of times that the wireless transmission is successful when rate r_i is selected until time slot t. The algorithm is presented as Algorithm 6.

Intuitively, in Algorithm 6, we maintain a pair of counters (i.e., $S_{i,1}(t)$ and $S_{i,2}(t)$) to track prediction and transmission outcomes when rate r_i is used, and use these counters to obtain the corresponding estimated probabilities for successful prediction and transmission. Different from the traditional KL-UCB with single feedback, we jointly consider the uncertainties in the estimated probabilities for successful prediction and transmission. It turns out that such a design yields the regret that asymptotically matches the lower bound we derived, as shown next.

Algorithm 6 KL-UCB with two-level feedback

Choose each rate once.
Subsequently, in each time slot t, select the rate $r_{i(t)}$ satisfying

$$r_{i(t)} = \arg\max_{i=1,2,\ldots,K} r_i \max \Bigg\{ pq : d\left(\frac{S_{i,1}(t)}{T_i(t)}, p\right) + d\left(\frac{S_{i,2}(t)}{T_i(t)}, q\right) \leq \frac{\log(1 + t\log^2 t)}{T_i(t)}; \frac{S_{i,1}(t)}{T_i(t)} \leq p \leq 1; \quad \frac{S_{i,2}(t)}{T_i(t)} \leq q \leq 1 \Bigg\}.$$

Theorem 4. *The regret achieved by Algorithm 6 for the stochastic multi-armed bandit problem with two-level feedback satisfies the following:*

$$\limsup_{n\to\infty} \frac{R(n)}{\log n} \leq \sum_{k\neq i^*} \frac{\Delta_k \mathbb{1}\{\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k} < 1\}}{\min\limits_{\substack{0\leq x,y\leq 1 \\ xy \geq \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_k}}} d(\alpha_k, x) + d(\beta_k, y)}.$$

Before diving deep into the technical details of the proof, we will present a brief overview.

3.4.1 Proof Overview

The intuition behind the proof technique we use is along similar lines as the intuition behind the proof technique of the standard KL-UCB algorithm (see [38] for more details). From the definition of regret in Equation (3.1), we note that in order to prove an upper bound on the regret achieved by Algorithm 6, we simply need to upper bound the number of times the algorithm transmits at a sub-optimal rate. To this end, we split the analysis into the following steps:

1. Let τ be defined as the time after which, for a small $\epsilon > 0$, the optimal rate's index (as computed by Algorithm 6) will always be strictly greater than $r_{i^*}(\alpha_{i^*} - \epsilon)(\beta_{i^*} - \epsilon)$. Intuitively, after time τ, the optimal rate's index will be close to its true value. We upper bound $\mathbb{E}[\tau]$ in Lemma 2.

2. We bound the expected number of times the index of a sub-optimal rate will be greater than $r_i\alpha_i\beta_i + \epsilon$, for a small $\epsilon > 0$ (see Lemma 3). After time τ, a sub-optimal rate will be transmitted only if its index exceeds its true index substantially, since the optimal rate's index will be close to its true index. Therefore, Lemma 3 allows us to upper bound the number of times a sub-optimal rate will be transmitted after the time τ.

3. We combine the above two results to bound the expected number of times a sub-optimal rate is transmitted and subsequently get the bound on regret.

3.4.2 Technical Details

Lemma 2. *(Underestimating the optimal arm) Let $X_1, X_2, ..., X_n$ and $Y_1, Y_2, ..., Y_n$ be independent and identically distributed Bernoulli random variables with mean α and β, respectively. Let $\hat{\alpha}_s = \frac{1}{s}\sum_{j=1}^{s} X_j$ and $\hat{\beta}_s = \frac{1}{s}\sum_{j=1}^{s} Y_j$. Let $\epsilon > 0$, $d^+(x,y) = d(x,y)\mathbb{I}(x \leq y)$, $f(t) = 1 + t\log^2 t$ and*

$$\tau = \min\Big\{t : \max_{1\leq s\leq n} d^+(\hat{\alpha}_s, \alpha - \epsilon) - \frac{\log f(t)}{s} \leq 0 \text{ and } \max_{1\leq s\leq n} d^+(\hat{\beta}_s, \beta - \epsilon) - \frac{\log f(t)}{s} \leq 0\Big\}$$

Then, $\mathbb{E}[\tau] \leq \frac{4}{\epsilon^2}$.

Proof.

$$\begin{aligned}
&\mathbb{P}(\tau > t)\\
\leq&\mathbb{P}(\{\exists 1 \leq s \leq n : d^+(\hat{\alpha}_s, \alpha - \epsilon) - \frac{\log f(t)}{s} > 0\} \text{ or}\\
&\quad \{\exists 1 \leq s \leq n : d^+(\hat{\beta}_s, \beta - \epsilon) - \frac{\log f(t)}{s} > 0\})\\
\leq&\mathbb{P}(\{\exists 1 \leq s \leq n : d^+(\hat{\alpha}_s, \alpha - \epsilon) - \frac{\log f(t)}{s} > 0\})\\
+&\mathbb{P}(\{\exists 1 \leq s \leq n : d^+(\hat{\beta}_s, \beta - \epsilon) - \frac{\log f(t)}{s} > 0\}),
\end{aligned} \tag{3.11}$$

where the last inequality follows from the union bound. Let us consider the first term on the right-hand side of the above inequality (the second term

can be analysed similarly). Note that the proof is similar to the proof of in [38, Lemma 10.7].

$$\begin{aligned}
&\mathbb{P}(\{\exists 1 \leq s \leq n : d^+(\hat{\alpha}_s, \alpha - \epsilon) - \frac{\log f(t)}{s} > 0\}) \\
&\leq \sum_{s=1}^{n} \mathbb{P}(d^+(\hat{\alpha}_s, \alpha - \epsilon) - \frac{\log f(t)}{s} > 0) \\
&\overset{(a)}{=} \sum_{s=1}^{n} \mathbb{P}(d(\hat{\alpha}_s, \alpha - \epsilon) - \frac{\log f(t)}{s} > 0, \hat{\alpha}_s < \alpha - \epsilon) \\
&\overset{(b)}{\leq} \sum_{s=1}^{n} \mathbb{P}(d(\hat{\alpha}_s, \alpha) > \frac{\log f(t)}{s} + 2\epsilon^2, \hat{\alpha}_s < \alpha) \\
&\overset{(c)}{\leq} \sum_{s=1}^{n} \exp(-s(\frac{\log f(t)}{s} + 2\epsilon^2)) \\
&= \frac{1}{f(t)} \sum_{s=1}^{n} \exp(-2\epsilon^2 s) \leq \frac{1}{2f(t)\epsilon^2},
\end{aligned} \tag{3.12}$$

where (a) follows from the definition of $d^+(x, y)$, (b) follows from [38, Lemma 10.2], (c) follows from [38, Corollary 10.4]. A similar analysis can be done for the second term on the right-hand side of Equation (3.11) to obtain:

$$\mathbb{P}(\{\exists 1 \leq s \leq n : d^+(\hat{\beta}_s, \beta - \epsilon) - \frac{\log f(t)}{s} > 0\}) \leq \frac{1}{2f(t)\epsilon^2}.$$

Combining the above inequality with Equations (3.11)-(3.12):

$$\mathbb{E}[\tau] = \int_0^\infty \mathbb{P}(\tau \geq t)dt \leq \int_0^\infty \frac{1}{f(t)\epsilon^2} dt \leq \frac{4}{\epsilon^2}, \tag{3.13}$$

where the last inequality follows from the fact that $\int_0^\infty \frac{1}{\log(1+t\log^2 t)} dt \leq 4$. □

Lemma 3. *(Overestimating a sub-optimal arm) Let $X_1, X_2, ..., X_n$ and $Y_1, Y_2, ..., Y_n$ be independent and identically distributed Bernoulli random variables with mean α and β, respectively. Assume that $\alpha\beta < 1$. Let $\hat{\alpha}_s = \frac{1}{s}\sum_{j=1}^{s} X_j$ and $\hat{\beta}_s = \frac{1}{s}\sum_{j=1}^{s} Y_j$. Let $h_c(x, y)$ be as defined in Lemma 1. Let*

$\Delta > 0$, *such that* $\alpha\beta + \Delta < 1$. *Let* $a > 0$ *and*

$$p^*, q^* = \arg \min_{\substack{0\le p,q\le 1 \\ pq \ge \alpha\beta+\Delta}} d(\alpha, p) + d(\beta, q),$$
$$\kappa = \sum_{s=1}^{n} \mathbb{I}\Big\{ \min_{\substack{0\le x,y\le 1 \\ xy \ge \alpha\beta+\Delta}} d(\hat{\alpha}_s, x) + d(\hat{\beta}_s, y) \le \frac{a}{s} \Big\}.$$

(1) If $0 \le \alpha, \beta < 1$, *then*

$$\mathbb{E}[\kappa] \le \inf_{\epsilon \in (0, \Gamma_{\alpha,\beta,\Delta})} \big(\frac{a}{M_{\alpha,\beta,\Delta,\epsilon}} + \frac{2}{\epsilon^2} \big),$$

where

$$M_{\alpha,\beta,\Delta,\epsilon} \triangleq d(\alpha + \epsilon, p^* - \frac{1 + \frac{4}{h_{\alpha\beta+\Delta}(\alpha,\beta)}}{1-\beta}\epsilon) + d(\beta + \epsilon, q^* - \frac{1 + \frac{4}{h_{\alpha\beta+\Delta}(\alpha,\beta)}}{1-\alpha}\epsilon),$$

and

$$\Gamma_{\alpha,\beta,\Delta} \triangleq \min\{ \frac{h^2_{\alpha\beta+\Delta}(\alpha,\beta)}{8}, \frac{\Delta}{1+\alpha+\beta}, \frac{(p^*-\alpha)(1-\beta)}{2-\beta+\frac{4}{h_{\alpha\beta+\Delta}(\alpha,\beta)}}, \frac{(q^*-\beta)(1-\alpha)}{2-\alpha+\frac{4}{h_{\alpha\beta+\Delta}(\alpha,\beta)}} \}.$$

(2) If $\alpha = 1$ *or* $\beta = 1$, *then*

$$\mathbb{E}[\kappa] \le \inf_{\epsilon \in (0,\Delta)} \big(\frac{a}{d(\gamma+\epsilon, \gamma+\Delta)} + \frac{3}{2\epsilon^2} \big),$$

where $\gamma \triangleq \min\{\alpha, \beta\}$.

Proof. Let $\epsilon \in (0, \Delta)$ and $\xi = \frac{a}{M_{\alpha,\beta,\Delta,\epsilon}}$. Let

$$\mathcal{L}_{s,\epsilon} \triangleq \big\{ \{X_i, Y_i\}_{i=1}^s : |\hat{\alpha}_s - \alpha| > \epsilon \text{ or } |\hat{\beta}_s - \beta| > \epsilon \big\}.$$

We have

$$
\begin{aligned}
&\mathbb{E}[\kappa] \\
&= \sum_{s=1}^{n} \mathbb{P}\big(\min_{\substack{0\le x,y\le 1;\\ xy\ge \alpha\beta+\Delta}} d(\hat{\alpha}_s, x) + d(\hat{\beta}_s, y) \le \frac{a}{s}\big) \\
&\le \sum_{s=1}^{n} \mathbb{P}(\mathcal{L}_{s,\epsilon}) + \sum_{s=1}^{n} \mathbb{P}\big(\min_{\substack{0\le x,y\le 1;\\ xy\ge \alpha\beta+\Delta}} d(\hat{\alpha}_s, x) + d(\hat{\beta}_s, y) \le \frac{a}{s} \Big| \mathcal{L}_{s,\epsilon}^{C}\big) \\
&\overset{(a)}{\le} \sum_{s=1}^{n} \mathbb{P}(\mathcal{L}_{s,\epsilon}) + \sum_{s=1}^{n} \mathbb{P}\left(M_{\alpha,\beta,\Delta,\epsilon} \le \frac{a}{s} \Big| \mathcal{L}_{s,\epsilon}^{C} \right) \\
&\overset{(b)}{\le} \sum_{s=1}^{n} \mathbb{P}(\mathcal{L}_{s,\epsilon}) + \xi \\
&\le \sum_{s=1}^{n} \mathbb{P}(|\hat{\alpha}_s - \alpha| > \epsilon) + \sum_{s=1}^{n} \mathbb{P}(|\hat{\beta}_s - \beta| > \epsilon) + \xi \\
&\overset{(c)}{\le} \sum_{s=1}^{\infty} \left(\exp\big(-sd(\alpha+\epsilon,\alpha)\big) + \exp\big(-sd(\alpha-\epsilon,\alpha)\big) \right) \\
&+ \sum_{s=1}^{\infty} \left(\exp\big(-sd(\beta+\epsilon,\beta)\big) + \exp\big(-sd(\beta-\epsilon,\beta)\big) \right) + \xi \\
&\le \frac{1}{d(\alpha+\epsilon,\alpha)} + \frac{1}{d(\alpha-\epsilon,\alpha)} + \frac{1}{d(\beta+\epsilon,\beta)} + \frac{1}{d(\beta-\epsilon,\beta)} + \xi \\
&\overset{(d)}{\le} \frac{2}{\epsilon^2} + \frac{a}{M_{\alpha,\beta,\Delta,\epsilon}},
\end{aligned}
$$

where (a) follows from the result 2(a) in Lemma 1, (b) follows from the definition of ξ, (c) follows from Chernoff's bound, and (d) follows from Pinsker's inequality. The result follows from taking the infimum over ϵ to obtain the tightest bound.

Let us consider the case when either $\alpha = 1$ or $\beta = 1$ (both cannot be equal to one due to the assumption that $\alpha\beta < 1$). Without loss of generality, let us assume that $\alpha = 1$ and $\beta < 1$. It can be readily seen that $p^* = 1$ and $q^* = \alpha\beta + \Delta = \beta + \Delta$ (similar to proof of Lemma 1). With the above observation, the result can be proved similar to the proof for the previous case (when $0 \le \alpha, \beta < 1$). □

Proof of Theorem 4:

Equipped with the three lemmas we have, we now prove the main result. Consider a sub-optimal rate r_i, i.e., $i \neq i^*$. Recall that $T_i(n)$ denotes the

number of times that the rate r_i is used for transmission until the end of time slot n. We will bound $\mathbb{E}[T_i(n)]$ and eventually bound the overall regret using Equation (3.1). Let $(p^*_{i^*}, q^*_{i^*}) = \arg \min_{\substack{0\le p,q\le 1 \\ pq\ge \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i}}} d(\alpha_{i^*}, p) + d(\beta_{i^*}, q)$ and $f(t) = 1 + t\log^2 t$. We will split the analysis into three cases: (i) $\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i} \le 1$, $0 \le \alpha_i, \beta_i < 1$, (ii) $\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i} \le 1$, either $\alpha_i = 1$ or $\beta_i = 1$, and (iii) $\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i} > 1$.

Case 1: $\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i} \le 1$, $0 \le \alpha_i, \beta_i < 1$.

Choose $\epsilon_1 > 0$ such that $\epsilon_1 r_{i^*}(\alpha_{i^*} + \beta_{i^*}) < r_{i^*}\alpha_{i^*}\beta_{i^*} - r_i\alpha_i\beta_i$. Also, let:

$$\tau = \min\Bigg\{t : \max_{1\le s\le n} d^+(\hat{\alpha}_{i^*,s}, \alpha_{i^*} - \epsilon_1) - \frac{\log f(t)}{s} \le 0 \\ \text{and } \max_{1\le s\le n} d^+(\hat{\beta}_{i^*,s}, \beta_{i^*} - \epsilon_1) - \frac{\log f(t)}{s} \le 0\Bigg\},$$

$$\kappa = \sum_{s=1}^{n} \mathbb{I}\Bigg\{\min_{\substack{0\le x,y\le 1; \\ xy\ge \Upsilon_{i,\epsilon_1}}} d(\hat{\alpha}_{i,s}, x) + d(\hat{\beta}_{i,s}, y) \le \frac{\log f(n)}{s}\Bigg\},$$

where $\Upsilon_{i,\epsilon_1} \triangleq \frac{r_{i^*}}{r_i}\big(\alpha_{i^*}\beta_{i^*} - \epsilon_1(\alpha_{i^*} + \beta_{i^*})\big)$.

We have:

$$\begin{aligned}
\mathbb{E}[T_i(n)] &= \mathbb{E}\big[\sum_{t=1}^{n} \mathbb{I}\{A_t = i\}\big] \\
&\le \mathbb{E}[\tau] + \mathbb{E}\big[\sum_{t=\tau+1}^{n} \mathbb{I}\{A_t = i\}\big] \\
&\overset{(a)}{\le} \mathbb{E}[\tau] + \mathbb{E}\big[\sum_{t=1}^{n} \mathbb{I}\{A_t = i \text{ and } \\
&\qquad\qquad \min_{\substack{0\le x,y\le 1; \\ xy\ge \Upsilon_{i,\epsilon_1}}} d(\hat{\alpha}_{i,s}, x) + d(\hat{\beta}_{i,s}, y) \le \frac{\log f(t)}{T_i(t-1)}\}\big] \\
&\overset{(b)}{\le} \mathbb{E}[\tau] + \mathbb{E}[\kappa],
\end{aligned}$$

where (a) follows from Algorithm 6 and the definition of τ, and (b) follows from the definition of κ. Combining the above inequality with different lemmas proved previously:

$$\begin{aligned}\mathbb{E}[T_i(n)] &\leq \frac{4}{\epsilon_1^2} + \inf_{\epsilon_2 \in (0, \Gamma_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1}})} \left(\frac{\log(1+t\log^2 t)}{M_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1},\epsilon_2}} + \frac{2}{\epsilon_2^2} \right) \\ &\leq \frac{4}{\epsilon_1^2} + \inf_{\epsilon_2 \in (0, \Gamma_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1}})} \left(\frac{\log(1+t\log^2 t)}{M'_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1},\epsilon_2}} + \frac{2}{\epsilon_2^2} \right),\end{aligned} \tag{3.14}$$

where the first inequality follows from Lemmas 2 and 3 with

$$\Delta_{i,\epsilon_1} = \frac{r_{i^*}\alpha_{i^*}\beta_{i^*} - r_i\alpha_i\beta_i - r_{i^*}\epsilon_1(\alpha_{i^*} + \beta_{i^*})}{r_i},$$

and $M_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1},\epsilon_2}$ and $\Gamma_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1}}$ as defined in Lemma 3.

The second inequality follows from Lemma 1 with

$$\begin{aligned}M'_{\alpha_i,\beta_i,\Delta_{i,\epsilon_1},\epsilon_2} &= d(\alpha_i + \epsilon_2, p_i^* - k_1\epsilon_1 - k_2\epsilon_2) \\ &\quad + d(\beta_i + \epsilon_2, q_i^* - k_3\epsilon_1 - k_4\epsilon_2),\end{aligned}$$

where

$$p_i^*, q_i^* = \arg \min_{\substack{0 \leq p,q \leq 1; \\ pq \geq \frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i}}} d(\alpha_i, p) + d(\beta_i, q),$$

$$k_1 = \frac{2(1-\alpha_i)r_{i^*}(\alpha_{i^*} + \beta_{i^*})}{h_{\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i}}(\alpha_i, \beta_i)}, k_2 = \frac{1 + \frac{4}{h_{\alpha_i\beta_i + \Delta_{i,\epsilon_1}}(\alpha_i,\beta_i)}}{1-\beta_i},$$

$$k_3 = \frac{2(1-\beta_i)r_{i^*}(\alpha_{i^*} + \beta_{i^*})}{h_{\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i}}(\alpha_i, \beta_i)}, k_4 = \frac{1 + \frac{4}{h_{\alpha_i\beta_i + \Delta_{i,\epsilon_1}}(\alpha_i,\beta_i)}}{1-\alpha_i}.$$

Case 2: $\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i} \leq 1$, either $\alpha_i = 1$ or $\beta_i < 1$. If either $\alpha_i = 1$ or $\beta_i = 1$, then we can use Lemma 3 and proceed as we did in the previous case to get:

$$\mathbb{E}[T_i(n)] \leq \frac{4}{\epsilon_1^2} + \inf_{\epsilon_2 \in (0, \Delta_{i,\epsilon_1})} \big(\frac{\log(1+t\log^2 t)}{d(\gamma + \epsilon_2, \gamma + \Delta_{i,\epsilon_1})} + \frac{3}{2\epsilon_2^2} \big), \tag{3.15}$$

where $\epsilon_1 \in (0, \frac{r_{i^*}\alpha_{i^*}\beta_{i^*} - r_i\alpha_i\beta_i}{r_{i^*}(\alpha_{i^*} + \beta_{i^*})})$, $\gamma = \min\{\alpha, \beta\}$ and Δ_{i,ϵ_1} as defined in the previous case.

Case 3: $\frac{r_{i^*}\alpha_{i^*}\beta_{i^*}}{r_i} > 1$. Choose $\epsilon > 0$ such that $r_i < r_{i^*}(\alpha_{i^*} - \epsilon)(\beta_{i^*} - \epsilon)$. Also, let:

$$\tau = \min\Big\{t : \max_{1\le s\le n} d^{+}(\hat{\alpha}_{i^*,s}, \alpha_{i^*} - \epsilon) - \frac{\log f(t)}{s} \le 0 \\ \text{and } \max_{1\le s\le n} d^{+}(\hat{\beta}_{i^*,s}, \beta_{i^*} - \epsilon) - \frac{\log f(t)}{s} \le 0\Big\}.$$

Note that after time τ, rate r_i will never be transmitted since $r_i < r_{i^*}(\alpha_{i^*} - \epsilon)(\beta_{i^*} - \epsilon)$. Therefore:

$$\mathbb{E}[T_i(n)] = \mathbb{E}\Big[\sum_{t=1}^{n} \mathbb{I}\{A_t = i\}\Big] \le \mathbb{E}[\tau] \le \frac{4}{\epsilon^2}, \tag{3.16}$$

where the last inequality follows from Lemma 2.

The final result can be obtained by combining Equations (3.14)–(3.16), using results 2 and 3 from Lemma 1 and taking limit superior (see [38, Chapter 8] for more details).

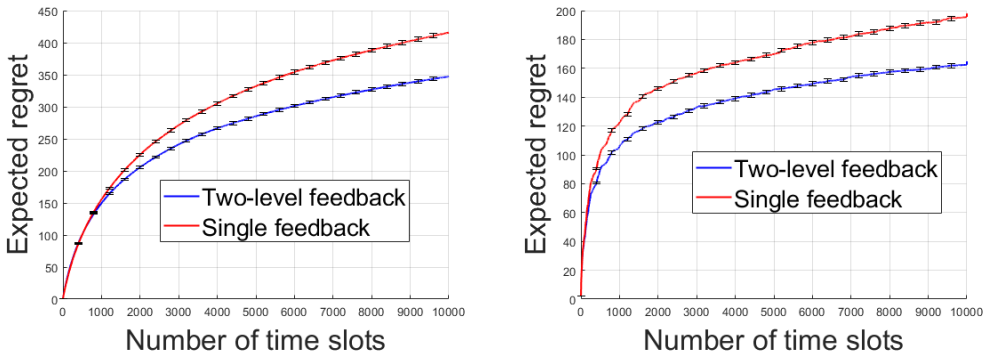
3.5 Experimental Results

In this section, we perform experiments to evaluate the regret performance of our proposed KL-UCB algorithm with two-level feedback information. We first consider a synthetic experiment with both the prediction and transmission outcomes being directly generated by Bernoulli random variables with means α_i and β_i, respectively, when rate r_i is used. In such a case, we consider the simulation setup with five different selected rates, as listed in Table 3.1. In the simulation setup, we run 5000 experiments to get the average results and each experiment's time horizon is set to 10^4 time slots. We plot the mean and 1.96 standard deviation (95% confidence interval) of the regret in Fig. 3.2a. We can observe from the Fig. 3.2a that KL-UCB algorithm with two-level feedback information outperforms its counterpart with single feedback information.

Next, we conduct an experiment using a real panoramic video trace from the dataset in [31]. We predict the FoV of the user by modeling the head motion as an autoregressive (AR) process and estimating the parameters of the AR process using the Yule-Walker equation [46]. Channel fading is not

Table 3.1: Synthetic simulation parameters

	arm1	arm2	arm3	arm4	arm5
Rate r_n	2	3	5	6	9
Prediction prob. α_n	0.1	0.3	0.5	0.65	0.9
Transmission prob. β_n	0.99	0.6	0.4	0.2	0.05
Average throughput	0.198	0.54	1	0.78	0.405



(a) Synthetic experiments (b) Real trace simulation

Figure 3.2: Regret performance

a part of this dataset, so we generate the channel state process synthetically. In particular, the channel rates are 2, 4, 5.5, 6, 7, and 8 with corresponding transmission probabilities 0.1, 0.15, 0.25, 0.25, 0.2, and 0.05. In our simulation, we consider five different transmission rates 1.5, 3.2, 5.2, 5.7, and 8.6. Different from the synthetic experiments, the optimal throughput is unknown and thus is obtained by running experiments for each fixed transmission rate. We plot the results in Fig. 3.2b. Observe that KL-UCB with two-level feedback still yields a smaller regret than that with single feedback in the real data trace.

CHAPTER 4

TWO TIME-SCALE TEMPORAL DIFFERENCE LEARNING ALGORITHMS

4.1 Background and Problem Formulation

A key component of reinforcement learning algorithms is to learn or approximate value functions under a given policy [47], [48], [49], [50], [51], [4]. Many existing algorithms for learning value functions are variants of the temporal-difference (TD) learning algorithms [47], [52], and can be viewed as stochastic approximation algorithms for minimizing the Bellman error (or objectives related to the Bellman error). Characterizing the convergence of these algorithms, such as TD(0), TD(λ), GTD, nonlinear GTD has been an important objective of reinforcement learning [49], [53], and [54]. The asymptotic convergence of these algorithms with diminishing steps has been established using stochastic approximation theory in many prior works (comprehensive surveys on stochastic approximations can be found in [55], [56], and [57]).

The conditions required for theoretically establishing asymptotic convergence in an algorithm with diminishing step-sizes imply that the learning rate becomes very small very quickly. As a result, the algorithm will require a very large number of samples to converge. Reinforcement learning algorithms used in practice follow a pre-determined learning rate (step-size) schedule which, in most cases, uses decaying step-sizes first and then a fixed step-size. This gap between theory and practice has prompted a sequence of works on the finite-time performance of temporal difference learning algorithms with either time-varying step-sizes or constant step-sizes [58, 59, 60, 61, 62, 63]. Most of these results are for single time-scale TD algorithms, except [59] which considers two time-scale algorithms with decaying step-sizes. Two time-scale TD algorithms are an important class of reinforcement learning algorithms because they can improve the convergence rate of TD learning or remedy the

instability of single time-scale TD in some cases. This chapter of the thesis focuses on two time-scale linear stochastic approximation algorithms with constant step-sizes. The model includes TDC, GTD and GTD2 as special cases (see [64], [65] and [49] for more details).

Besides the theoretical analysis of the finite-time performance of two time-scale reinforcement learning algorithms, another important aspect of reinforcement learning algorithms, which is imperative in practice but has been largely overlooked, is the design of learning rate schedule, i.e., how to choose proper step-sizes to improve the sample efficiency and reduce the learning time. This report addresses this important question by developing principled heuristics based on the finite-time performance bounds.

The main contributions of this chapter are summarized below.

1. **Finite Time Performance Bounds:** We study two time-scale linear stochastic approximation algorithms, driven by Markovian samples. We establish finite time bounds on the mean-square error with respect to the fixed point of the corresponding ordinary differential equations (ODEs). The performance bound consists of two parts: a steady-state error and a transient error, where the steady-state error is determined by the step-sizes but independent of the number of samples (or number of iterations), and the transient error depends on both step-sizes and the number of samples. The transient error decays geometrically as the number of samples increases. The key differences between our contributions and [59] include (i) we do not require a sparse projection step in the algorithm; and (ii) we assume constant step-sizes which allows us to develop the adaptive step-size selection heuristic mentioned next.

2. **Adaptive Learning Rate Selection:** Based on the finite-time performance bounds, in particular, the steady-state error and the transient error terms in the bounds, we propose an adaptive learning rate selection scheme. The intuition is to use a constant learning rate until the transient error is dominated by the steady-state error; after that, running the algorithm further with the same learning rate is not very useful and therefore, we reduce the learning rate at this time. To apply adaptive learning rate selection in a model-free fashion, we develop data-driven heuristics to determine the time at which the transient error is close to the steady-state error. A useful property of our adaptive

rate selection scheme is that it can be used with any learning rate schedule which already exists in many machine learning software platforms: one can start with the initial learning rate suggested by such schedules and get improved performance by using our adaptive scheme. Our experiments on Mountain Car and Inverted Pendulum show that our adaptive learning rate selection significantly improves the convergence rates as compared to optimal polynomial decay learning rate strategies (see [59] and [66] for more details on polynomial decay step-size rules).

4.1.1 Model, Notation and Assumptions

We consider the following two time-scale linear stochastic approximation algorithm:

$$\begin{aligned} U_{k+1} &= U_k + \epsilon^{\alpha} \left(A_{uu}(X_k) U_k + A_{uv}(X_k) V_k + b_u(X_k) \right) \\ V_{k+1} &= V_k + \epsilon^{\beta} \left(A_{vu}(X_k) U_k + A_{vv}(X_k) V_k + b_v(X_k) \right), \end{aligned} \tag{4.1}$$

where $\{X_k\}$ are the samples from a Markov process. We assume $\beta < \alpha$ so that, over $\epsilon^{-\beta}$ iterations, the change in V is $O(1)$ while the change in U is $O\left(\epsilon^{\alpha-\beta}\right)$. Therefore, V is updated at a faster time scale than U.

In the context of reinforcement learning, when combined with linear function approximation of the value function, GTD, GTD2, and and TDC can be viewed as two time-scale linear stochastic approximation algorithms, and can be described in the same form as in Equation (4.1). For example, TDC with linear function approximation is as follows:

$$\begin{aligned} U_{k+1} =& U_k + \epsilon^{\alpha} \left(\phi(X_k) - \zeta \phi(X_{k+1}) \right) \phi^{\top}(X_k) V_k \\ V_{k+1} =& V_k + \epsilon^{\beta} \left(\delta_k - \phi^{\top}(X_k) V_k \right) \phi(X_k), \end{aligned}$$

where ζ is the discount factor, $\phi(x)$ is the feature vector of state x, U_k is the weight vector such that $\phi^{\top}(x)U_k$ is the approximation of value function of state x at iteration k, $\delta_k = c(X_k) + \zeta \phi^{\top}(X_{k+1})U_k - \phi^{\top}(X_k)U_k$ is the TD error, and V_k is the weight vector such that $\phi^{\top}(x)V_k$ is the estimate of the TD error for state x at iteration k.

We now summarize the notation we use throughout in this report and the assumptions we make.

- **Assumption 1:** $\{X_k\}$ is a Markov chain with state space $\mathcal{S}$. We assume that the following two limits exist:

$$\begin{pmatrix} \bar{A}_{uu} & \bar{A}_{uv} \\ \bar{A}_{vu} & \bar{A}_{vv} \end{pmatrix} = \lim_{k\to\infty} \begin{pmatrix} \mathbb{E}\left[A_{uu}(X_k)\right] & \mathbb{E}\left[A_{uv}(X_k)\right] \\ \mathbb{E}\left[A_{vu}(X_k)\right] & \mathbb{E}\left[A_{vv}(X_k)\right] \end{pmatrix},$$
$$\begin{pmatrix} \bar{b}_u & \bar{b}_v \end{pmatrix} = \lim_{k\to\infty} \begin{pmatrix} \mathbb{E}[b_u(X_k)] & \mathbb{E}[b_v(X_k)] \end{pmatrix} = 0.$$

  Note that without the loss of generality, we assume $\bar{b} = 0$ which allows for the fixed point of the associated ODEs to be 0. This can be guaranteed by appropriate centering. We define

$$\begin{aligned} B(X_k) =& A_{uu}(X_k) - A_{uv}(X_k)\bar{A}_{vv}^{-1}\bar{A}_{vu}, \qquad & \bar{B} =& \bar{A}_{uu} - \bar{A}_{uv}\bar{A}_{vv}^{-1}A_{vu}, \\ \tilde{B}(X_k) =& A_{vu}(X_k) - A_{vv}(X_k)\bar{A}_{vv}^{-1}\bar{A}_{vu}, \qquad & \bar{\tilde{B}} =& \bar{A}_{vu} - \bar{A}_{vv}\bar{A}_{vv}^{-1}\bar{A}_{vu}. \end{aligned}$$

- **Assumption 2:** We assume the following:

$$\begin{aligned} \max\{\|b_u(x)\|, \|b_v(x)\|\} \leq b_{\max} < \infty \forall x \in \mathcal{S}, \\ \max\{\|B(x)\|, \|\tilde{B}(x)\|, \|A_{uu}(x)\|, \|A_{vu}(x)\|, \\ \|A_{uv}(x)\|, \|A_{vv}(x)\|\} \leq 1 \forall x \in \mathcal{S}. \end{aligned}$$

  Note that the above assumptions imply that the steady-state limits of the random matrices/vectors will also satisfy the same inequalities.

- **Assumption 3:** We assume $\bar{A}_{vv}$ and $\bar{B}$ are Hurwitz and $\bar{A}_{vv}$ is invertible. Let P_u and P_v be the solutions to the following Lyapunov equations:

$$\begin{aligned} -I &= \bar{B}^\top P_u + P_u \bar{B}, \\ -I &= \bar{A}_{vv}^\top P_v + P_v \bar{A}_{vv}. \end{aligned}$$

  Since both $\bar{A}_{vv}$ and $\bar{B}$ are Hurwitz, P_u and P_v are real positive definite matrices.

- **Assumption 4:** Define $\tau_\Delta \geq 1$ to be the mixing time of the Markov

chain $\{X_k\}$. We assume

$$\begin{aligned}
\|\mathbb{E}[b_k|X_0 = i]\| &\leq \Delta, \forall i, \forall k \geq \tau_\Delta, \\
\|\bar{B} - \mathbb{E}[B(X_k)|X_0 = i]\| &\leq \Delta, \forall i, \forall k \geq \tau_\Delta, \\
\|\bar{\tilde{B}} - \mathbb{E}[\tilde{B}(X_k)|X_0 = i]\| &\leq \Delta, \forall i, \forall k \geq \tau_\Delta, \\
\|\bar{A}_{uv} - \mathbb{E}[A_{uv}(X_k)|X_0 = i]\| &\leq \Delta, \forall i, \forall k \geq \tau_\Delta, \\
\|\bar{A}_{vv} - \mathbb{E}[A_{vv}(X_k)|X_0 = i]\| &\leq \Delta, \forall i, \forall k \geq \tau_\Delta.
\end{aligned}$$

- **Assumption 5:** As in [63], we assume that there exists $K \geq 1$ such that $\tau_\Delta \leq K \log(\frac{1}{\Delta})$. For convenience, we choose

$$\Delta = 2\epsilon^\alpha \left(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\| + \epsilon^{\beta-\alpha}\right)$$

  and drop the subscript from τ_Δ, i.e., $\tau_\Delta = \tau$. Also, for convenience, we assume that ϵ is small enough such that $\tilde{\epsilon}\tau \leq \frac{1}{4}$, where $\tilde{\epsilon} = \Delta = 2\epsilon^\alpha \left(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\| + \epsilon^{\beta-\alpha}\right)$.

We further define the following notation:

- Define matrix

$$P = \begin{pmatrix} \frac{\xi_v}{\xi_u+\xi_v} P_u & 0 \\ 0 & \frac{\xi_u}{\xi_u+\xi_v} P_v \end{pmatrix}, \tag{4.2}$$

  where $\xi_u = 2\|P_u\bar{A}_{uv}\|$ and $\xi_v = 2\left\|P_v\bar{A}_{vv}^{-1}\bar{A}_{vu}\bar{B}\right\|$.

- Let $\gamma_{\max}$ and $\gamma_{\min}$ denote the largest and smallest eigenvalues of P_u and P_v, respectively. So $\gamma_{\max}$ and $\gamma_{\min}$ are also upper and lower bounds on the eigenvalues of P.

4.2 Finite-Time Performance Bounds

To establish the finite-time performance guarantees of the two time-scale linear stochastic approximation algorithm (in Equation (4.1)), we define

$$Z_k = V_k + \bar{A}_{vv}^{-1}\bar{A}_{vu}U_k \quad \text{and} \quad \Theta_k = \begin{pmatrix} U_k \\ Z_k \end{pmatrix}.$$

Then we consider the following Lyapunov function:

$$W(\Theta_k) = \Theta_k^\top P \Theta_k, \tag{4.3}$$

where P is a symmetric positive definite matrix defined in Equation (4.2) (P is positive definite because both P_u and P_v are positive definite matrices). The reason to introduce Z_k will become clear when we introduce the key idea of our analysis based on singular perturbation theory.

The following lemma bounds the expected change in the Lyapunov function in one time step.

Lemma 4. *For any $k \geq \tau$ and ϵ, α, and β such that $\eta_1 \tilde{\epsilon} \tau + 2\frac{\tilde{\epsilon}^2}{\epsilon^\alpha}\gamma_{\max} \leq \frac{\kappa_1}{2}$, the following inequality holds:*

$$\mathbb{E}[W(\Theta_{k+1}) - W(\Theta_k)] \leq -\frac{\epsilon^\alpha}{\gamma_{\max}}\left(\frac{\kappa_1}{2} - \kappa_2 \epsilon^{\alpha-\beta}\right)\mathbb{E}[W(\Theta_k)] + \epsilon^{2\beta}\tau\eta_2,$$

where $\tilde{\epsilon} = 2\epsilon^\alpha\left(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\| + \epsilon^{\beta-\alpha}\right)$, and η_1, η_2 κ_1, and κ_2 are constants independent of ϵ.

The proof of Lemma 4 is somewhat involved, and we defer the proof to the supplementary material section. The definitions of η_1, η_2, κ_1 and κ_2 are also not presented here. Here, we provide some intuition behind the result by studying a related ordinary differential equation (ODE). In particular, consider the expected change in the stochastic system divided by the slow time-scale step-size ϵ^α:

$$\begin{aligned}
&\frac{\mathbb{E}[U_{k+1} - U_k | U_{k-\tau} = u, V_{k-\tau} = v, X_{k-\tau} = x]}{\epsilon^\alpha} \\
=&\mathbb{E}\left[\left(A_{uu}(X_k)U_k + A_{uv}(X_k)V_k + b_u\right) \middle| U_{k-\tau} = u, V_{k-\tau} = v, X_{k-\tau} = x\right] \\
&\epsilon^{\alpha-\beta}\frac{\mathbb{E}[V_{k+1} - V_k | U_{k-\tau} = u, V_{k-\tau} = v, X_{k-\tau} = x]}{\epsilon^\alpha} \\
=&\mathbb{E}\left[\left(A_{vu}(X_k)U_k + A_{vv}(X_k)V_k + b_v(X_k)\right) \middle| U_{k-\tau} = u, V_{k-\tau} = v, X_{k-\tau} = x\right],
\end{aligned} \tag{4.4}$$

where the expectation is conditioned sufficiently in the past in terms of the underlying Markov chain (i.e. conditioned on the state at time $k - \tau$ instead of k) so the expectation is approximately in steady-state.

Approximating the left-hand side by derivatives and the right-hand side

using steady-state expectations, we get the following ODEs:

$$\dot{u} = \bar{A}_{uu}u + \bar{A}_{uv}v \tag{4.5}$$

$$\epsilon^{\alpha-\beta}\dot{v} = \bar{A}_{vu}u + \bar{A}_{vv}v. \tag{4.6}$$

Note that, in the limit as $\epsilon \to 0$, the second of the above two ODEs becomes an algebraic equation, instead of a differential equation. In the control theory literature, such systems are called singularly-perturbed differential equations, see for example [67]. In [68, Chapter 11], the following Lyapunov equation has been suggested to study the stability of such singularly perturbed ODEs (see the supplementary section for more details):

$$W(u,v) = du^\top P_u u + (1-d)\left(v + \bar{A}_{vv}^{-1}\bar{A}_{vu}u\right)^\top P_v \left(v + \bar{A}_{vv}^{-1}\bar{A}_{vu}u\right), \tag{4.7}$$

for $d \in [0,1]$. The function W mentioned earlier in Equation (4.3) is the same as above for a carefully chosen d.

The intuition behind the result in Lemma 4 can be understood by studying the dynamics of the above Lyapunov function in the ODE setting. To simplify the notation, we define $z = v + \bar{A}_{vv}^{-1}\bar{A}_{vu}u$, so the Lyapunov function can also be written as

$$W(u,z) = du^\top P_u u + (1-d)z^\top P_v z, \tag{4.8}$$

and adapting the manipulations for nonlinear ODEs in [68, Chapter 11] to our linear model, we get

$$\begin{aligned} \dot{W} =& 2du^T P_u \dot{u} + 2(1-d)z^\top P_v \dot{z} \\ \leq& -\begin{pmatrix} \|u\| & \|z\| \end{pmatrix} \tilde{\Psi} \begin{pmatrix} \|u\| \\ \|z\| \end{pmatrix}, \end{aligned} \tag{4.9}$$

where

$$\tilde{\Psi} = \begin{pmatrix} d & -d\gamma_{\max} - (1-d)\gamma_{\max}\sigma_{\min} \\ -d\gamma_{\max} - (1-d)\gamma_{\max}\sigma_{\min} & \left(\frac{1-d}{2\epsilon^{\alpha-\beta}} - (1-d)\gamma_{\max}\sigma_{\min}\right) \end{pmatrix}, \tag{4.10}$$

and $\sigma_{\min}$ is the upper bound on the induced 2-norm of the inverse of matrices $\bar{A}_{uu}, \bar{A}_{uv}, \bar{A}_{vv}$ and $\bar{A}_{vu}$. Note that $\tilde{\Psi}$ is positive definite when

$$d\left(\frac{1-d}{2\epsilon^{\alpha-\beta}} - (1-d)\gamma_{\max}\sigma_{\min}\right) \geq \left(d\gamma_{\max} + (1-d)\gamma_{\max}\sigma_{\min}\right)^2, \tag{4.11}$$

i.e., when

$$\epsilon^{\alpha-\beta} \leq \frac{d(1-d)}{2d(1-d)\gamma_{\max}\sigma_{\min} + \left(d\gamma_{\max} + (1-d)\gamma_{\max}\sigma_{\min}\right)^2}. \tag{4.12}$$

Let $\tilde{\lambda}_{\min}$ denote the smallest eigenvalue of $\tilde{\Psi}$. We have

$$\dot{W} \leq -\tilde{\lambda}_{\min}\left(\|u\|^2 + \|z\|^2\right) \leq -\frac{\tilde{\lambda}_{\min}}{\gamma_{\max}} W. \tag{4.13}$$

In particular, recall that we obtained the ODEs by dividing by the step-size ϵ^α. Therefore, for the discrete equations, we would expect

$$\mathbb{E}[W(\Theta_{k+1}) - W(\Theta_k)] \approx\leq -\epsilon^\alpha \frac{\tilde{\lambda}_{\min}}{\gamma_{\max}} \mathbb{E}\left[W(\Theta_k)\right], \tag{4.14}$$

which resembles the transient term of the upper bound in Lemma 4. The exact expression in the discrete, stochastic case is of course different and additionally includes a steady-state term, which is not captured by the ODE analysis above.

Now, we are ready to the state the main theorem.

Theorem 5. *For any $k \geq \tau$, ϵ, α and β such that $\eta_1\tilde{\epsilon}\tau + 2\frac{\tilde{\epsilon}^2}{\epsilon^\alpha}\gamma_{\max} \leq \frac{\kappa_1}{2}$, we have*

$$\begin{aligned}\mathbb{E}[\|\Theta_k\|^2] \leq & \frac{\gamma_{\max}}{\gamma_{\min}}\left(1 - \frac{\epsilon^\alpha}{\gamma_{\max}}\left(\frac{\kappa_1}{2} - \kappa_2\epsilon^{\alpha-\beta}\right)\right)^{k-\tau}\left(1.5\|\Theta_0\| + 0.5b_{\max}\right)^2 \\ & + \epsilon^{2\beta-\alpha}\frac{\gamma_{\max}}{\gamma_{\min}}\frac{\eta_2\tau}{\left(\frac{\kappa_1}{2} - \kappa_2\epsilon^{\alpha-\beta}\right)}.\end{aligned}$$

Theorem 5 essentially states that the expected error for a two-time scale linear stochastic approximation algorithm comprises two terms: a *transient error* term which decays geometrically with time and a *steady-state error* term which is directly proportional to $\epsilon^{2\beta-\alpha}$ and the mixing time. This characterization of the finite-time error is useful in understanding the impact of different algorithmic and problem parameters on the rate of convergence, allowing the design of efficient techniques such as the adaptive learning rate rule which we will present in the next section.

4.3 Adaptive Selection of Learning Rates

Equipped with the theoretical results from the previous section, one interesting question that arises is the following: *given a time-scale ratio $\lambda = \frac{\alpha}{\beta}$, can we use the finite-time performance bound to design a rule for adapting the learning rate to optimize performance?*

In order to simplify the discussion, let $\epsilon^\beta = \mu$ and $\epsilon^\alpha = \mu^\lambda$. Therefore, Theorem 5 can be simplified and written as

$$\mathbb{E}[\|\Theta_k\|^2] \leq K_1 \left(1 - \mu^\lambda \left(\frac{\kappa_1}{2\gamma_{\max}} - \frac{\kappa_2}{\gamma_{\max}}\mu^{\lambda-1}\right)\right)^k + \mu^{2-\lambda}\frac{K_2}{\left(\frac{\kappa_1}{2} - \kappa_2\mu^{\lambda-1}\right)}, \tag{4.15}$$

where K_1 and K_2 are problem-dependent positive constants. Since we want the system to be stable, we will assume that μ is small enough such that $\frac{\kappa_1}{2\gamma_{\max}} - \frac{\kappa_2}{\gamma_{\max}}\mu^{\lambda-1} = c > 0$. Plugging this condition in Equation (4.15), we get

$$\mathbb{E}[\|\Theta_k\|^2] \leq K_1 \left(1 - c\mu^\lambda\right)^k + \frac{K_2\mu^{2-\lambda}}{\gamma_{\max}c}. \tag{4.16}$$

In order to optimize performance for a given number of samples, we would like to choose the learning rate μ as a function of the time step. In principle, one can assume time-varying learning rates, derive more general mean-squared error expressions (similar to Theorem 5), and then try to optimize over the learning rates to minimize the error for a given number of samples. However, this optimization problem is computationally intractable. We note that even if we assume that we are only going to change the learning rate a finite number of times, the resulting optimization problem of finding the times at which such changes are performed and finding the learning rate at these change points is an equally intractable optimization problem. Therefore, we have to devise simpler adaptive learning rate rules.

To motivate our learning rate rule, we first consider a time T such that errors due to the transient and steady-state parts in Equation (4.16) are equal, i.e.,

$$K_1(1 - c\mu^\lambda)^T = \frac{K_2\mu^{2-\lambda}}{\gamma_{\max}c}. \tag{4.17}$$

From this time onwards, running the two time-scale stochastic approximation algorithm any further with μ as the learning rate is not going to significantly

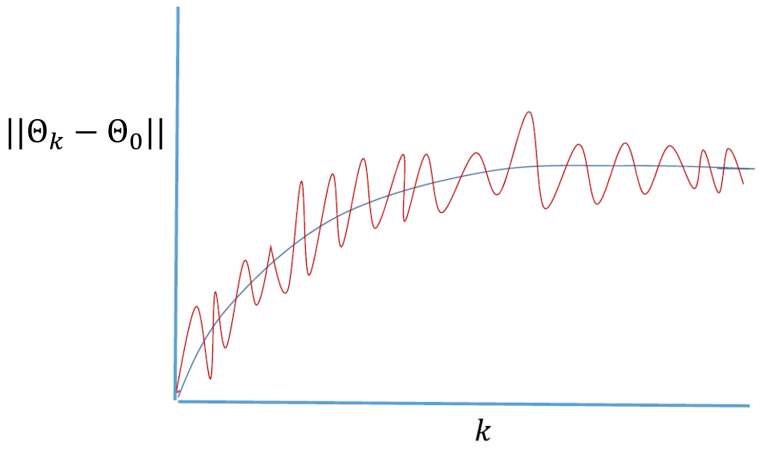


Figure 4.1: The evolution of $\|\Theta_k - \Theta_0\|$.

improve the mean-squared error. In particular, the mean-squared error beyond this time is upper bounded by twice the steady-state error $\frac{K_2\mu^{2-\lambda}}{\gamma_{\max}c}$. Thus, at time T, it makes sense to reset μ as $\mu \leftarrow \mu/\xi$, where $\xi > 1$ is a hyperparameter. Roughly speaking, T is the time at which one is close to steady-state for a given learning rate, and therefore, it is the time to reduce the learning rate to get to a new "steady-state" with a smaller error.

The key difficulty in implementing the above idea is that it is difficult to determine T. For ease of exposition, we considered a system centered around 0 in our analysis (i.e., $\Theta^* = 0$). More generally, the results presented in Theorem 5 and Equations (4.15) - (4.16) will have Θ_k replaced by $\Theta_k - \Theta^*$. In any practical application, Θ^* will be unknown. Thus, we cannot determine $\|\Theta_k - \Theta^*\|$ as a function of k and hence, it is difficult to use this approach.

Our idea to overcome this difficulty is to estimate whether the algorithm is close to its steady-state by observing $\|\Theta_k - \Theta_0\|$ where Θ_0 is our initial guess for the unknown parameter vector and is thus known to us. Note that $\|\Theta_k - \Theta_0\|$ is zero at $k = 0$ and will increase (with some fluctuations due to randomness) to $\|\Theta^* - \Theta_0\|$ in steady-state (see Fig. 4.1 for an illustration). Roughly speaking, we approximate the curve in this figure by a sequence of straight lines, i.e., perform a piecewise linear approximation, and conclude that the system has reached steady-state when the lines become approximately horizontal. We provide the details next.

To derive a test to estimate whether $\|\Theta_k - \Theta_0\|$ has reached steady-state, we first note the following inequality for $k \geq T$ (i.e., after the steady-state time defined in Equation (4.17)):

$$\begin{aligned}\mathbb{E}[\|\Theta_0 - \Theta^*\|] - \mathbb{E}[\|\Theta_k - \Theta^*\|] \leq & \mathbb{E}[\|\Theta_k - \Theta_0\|] \leq \mathbb{E}[\|\Theta_k - \Theta^*\|] \\ & + \mathbb{E}[\|\Theta_0 - \Theta^*\|] \\ \Rightarrow d - \sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max}c}} \leq & \mathbb{E}[\|\Theta_k - \Theta_0\|] \leq d + \sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max}c}},\end{aligned} \tag{4.18}$$

where the first pair of inequalities follow from the triangle inequality and the second pair of inequalities follow from Equations (4.16) - (4.17), Jensen's inequality and letting $d = \mathbb{E}[\|\Theta_0 - \Theta^*\|]$. Now, for $k \geq T$, consider the following N points: $\{X_i = i, Y_i = \|\Theta_{k+i} - \Theta_0\|\}_{i=1}^{N}$. Since these points are all obtained after the "steady-state" is reached, if we draw the best-fit line through these points, its slope should be small. More precisely, let ψ_N denote the slope of the best-fit line passing through these N points. Using Equation (4.18) along with formulas for the slope in linear regression, and after some algebraic manipulations, one can show that (details in the supplementary material section):

$$|\mathbb{E}[\psi_N]| = O\left(\frac{\mu^{1-\frac{\lambda}{2}}}{N}\right), \quad \text{Var}(\psi_N) = O\left(\frac{1}{N^2}\right). \tag{4.19}$$

Therefore, if $N \geq \frac{\chi}{\mu^{\frac{\lambda}{2}}}$, then the slope of the best-fit line connecting $\{X_i, Y_i\}$ will be $O\left(\frac{\mu^{1-\frac{\lambda}{2}}}{N}\right)$ with high probability (for a sufficiently large constant $\chi > 0$). On the other hand, when the algorithm is in the transient state, the difference between $\|\Theta_{k+m} - \Theta_0\|$ and $\|\Theta_k - \Theta_0\|$ will be $O(m\mu)$ since Θ_k changes by $O(\mu)$ from one time slot to the next. Using this fact, the slope of the best-fit line through N consecutive points in the transient state can be shown to be $O(\mu)$, similar to Equation (4.19). Since we choose $N \geq \frac{\chi}{\mu^{\frac{\lambda}{2}}}$, the slope of the best-fit line in steady state, i.e., $O\left(\frac{\mu^{1-\frac{\lambda}{2}}}{N}\right)$ will be lower than the slope of the best-fit line in the transient phase, i.e., $O(\mu)$ (for a sufficiently large χ). We use this fact as a diagnostic test to determine whether or not the algorithm has entered steady-state. If the diagnostic test returns true, we update the learning rate (see Algorithm 7).

Algorithm 7 Adaptive learning rate rule

Hyperparameters: ρ, σ, ξ, N.
Initialize $\mu = \rho$, $\psi_N = 2\sigma\mu^{1-\frac{\lambda}{2}}$, Θ_0, $\Theta_{\text{ini}} = \Theta_0$.
for $i = 1, 2, ...$ **do**
  Do two time-scale algorithm update.
  Compute $\psi_N = \text{Slope}\left(\{k, \|\Theta_{i-k} - \Theta_{\text{ini}}\|\}_{k=0}^{N-1}\right)$.
  if $\psi_N < \frac{\sigma\mu^{1-\frac{\lambda}{2}}}{N}$ **then**
    $\mu = \frac{\mu}{\xi}$.
    $\Theta_{\text{ini}} = \Theta_i$.
  end if
end for

We note that our adaptive learning rate rule will also work for single time-scale reinforcement learning algorithms such as TD(0) since our expressions for the mean-square error, when specialized to the case of a single time-scale, will recover the result in [63] (see [69] for more details).

4.4 Experiments

We implemented our adaptive learning rate schedule on two popular classic control problems in reinforcement learning - Mountain Car and Inverted Pendulum, and compared its performance with the optimal polynomial decay learning rate rule suggested in [59] (described in the next subsection). We evaluated the following policies using the two time-scale TDC algorithm (see [65] for more details regarding TDC):

- Mountain Car - At each time step, choose a random action $\in \{0, 2\}$, i.e., accelerate randomly to the left or right.
- Inverted Pendulum - At each time step, choose a random action in the entire action space, i.e., apply a random torque $\in [-2.0, 2.0]$ at the pivot point.

Since the true value of Θ^* is not known in both the problems we consider, to quantify the performance of the TDC algorithm, we used the error metric known as the *norm of the expected TD update* (NEU, see [65] for more

details). For both problems, we used a $O(3)$ Fourier basis (see [70] for more details) to approximate the value function and used 0.95 as the discount factor. More details regarding the experimental setup can be found in the supplementary section.

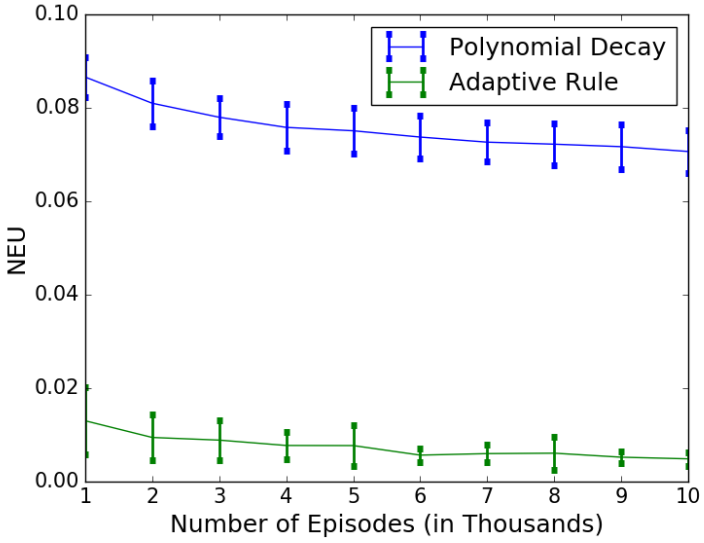


Figure 4.2: Mountain Car

4.4.1 Learning Rate Rules and Tuning

1. The optimal polynomial decay rule suggested in [59] is the following: at time step k, choose $\epsilon_k^\alpha = \frac{1}{(k+1)^\alpha}$ and $\epsilon_k^\beta = \frac{1}{(k+1)^\beta}$, where $\alpha \to 1$ and $\beta \to \frac{2}{3}$. For our experiments, we chose $\alpha = 0.99$ and $\beta = 0.66$. This implies $\lambda = \frac{\alpha}{\beta} = 1.5$. Since the problems we considered require smaller initial step-sizes for convergence, we let $\epsilon_k^\alpha = \frac{\rho_0}{(k+1)^\alpha}$ and $\epsilon_k^\beta = \frac{\rho_0}{(k+1)^\beta}$ and did a grid search to determine the best ρ_0, i.e., the best initial learning rate. The following values for ρ_0 were found to be the best: *Mountain Car* - $\rho_0 = 0.2$, *Inverted Pendulum* - $\rho_0 = 0.2$.

2. For our proposed adaptive learning rate rule, we fixed $\xi = 1.2, N = 200$ in both problems since we did not want the decay in the learning rate to be too aggressive and the resource consumption for slope computation to be high. We also set $\lambda = 1.5$ as in the polynomial decay case to have a fair comparison. We then fixed ρ and conducted a grid search

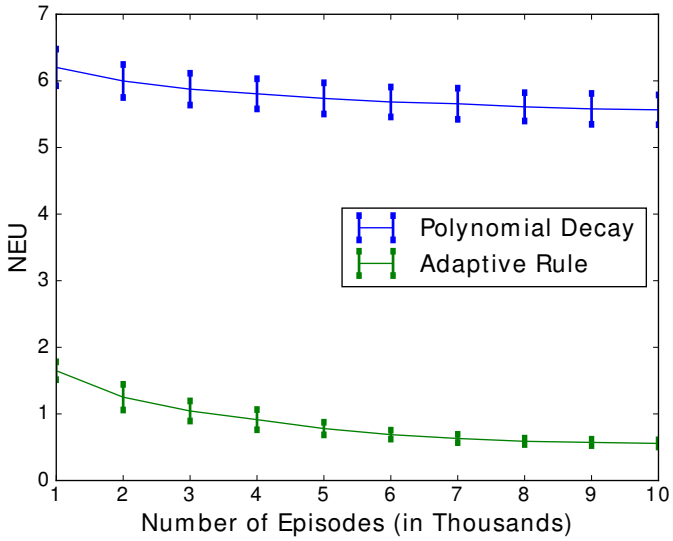


Figure 4.3: Inverted Pendulum

to find the best σ. Subsequently, we conducted a grid search over ρ. Interestingly, the adaptive learning rate rule was reasonably robust to the value of ρ. We used $\rho = 0.05$ in Inverted Pendulum and $\rho = 0.1$ in Mountain Car. Effectively, the only hyperparameter that affected the rule's performance significantly was σ. The following values for σ were found to be the best: *Mountain Car* - $\sigma = 0.001$, *Inverted Pendulum* - $\sigma = 0.01$.

4.4.2 Results

For each experiment, one run involved the following: $10,000$ episodes with the number of iterations in each episode being 50 and 200 for Inverted Pendulum and Mountain Car respectively. After every $1,000$ episodes, training/learning was paused and the NEU was computed by averaging over $1,000$ test episodes. We initialized $\Theta_0 = 0$. For Mountain Car, 50 such runs were conducted and the results were computed by averaging over these runs. For Inverted Pendulum, 100 runs were conducted and the results were computed by averaging over these runs. Note that the learning rate for each adaptive strategy was adapted at the episodic level due to the episodic nature of the problems. The results are reported in Figs. 4.2 and 4.3. As is clear from the figures, our proposed adaptive learning rate rule significantly outperforms

the optimal polynomial decay rule.

4.5 Supplementary Material

4.5.1 Proof of Lemma 4

The proof proceeds along similar lines as the corresponding proof in [63]. However, the results there cannot be directly applied to get the bounds in this paper due to the fact that we would like to separate out the effects of the ϵ, α and β from the other problem parameters, and additionally, the Lyapunov function used here is different.

Recall that

$$Z_k = V_k + \bar{A}_{vv}^{-1}\bar{A}_{vu}U_k,$$

so the stochastic recursions in terms of (U, Z) are

$$\begin{aligned}
U_{k+1} &= U_k + \epsilon^\alpha \left(B(X_k)U_k + A_{uv}(X_k)Z_k + b_u(X_k)\right) \\
Z_{k+1} &= Z_k + \bar{A}_{22}^{-1}\bar{A}_{21}(U_{k+1} - U_k) + \epsilon^\beta \left(\tilde{B}(X_k)U_k + A_{vv}(X_k)Z_k + b_v(X_k)\right) \\
&= Z_k + \epsilon^\alpha \bar{A}_{22}^{-1}\bar{A}_{21}\left(B(X_k)U_k + A_{uv}(X_k)Z_k + b_u(X_k)\right) \\
&\qquad + \epsilon^\beta \left(\tilde{B}(X_k)U_k + A_{vv}(X_k)Z_k + b_v(X_k)\right),
\end{aligned}$$

which can be written as a stochastic recursion in terms of $\Theta_k = (U_k, Z_k)$ as follows

$$\Theta_{k+1} = \Theta_k + \epsilon^\alpha \left(\tilde{A}(X_k) + \tilde{b}(X_k)\right), \tag{4.20}$$

where

$$\tilde{A}(X_k) = \begin{pmatrix} B(X_k) & A_{uv}(X_k) \\ \bar{A}_{vv}^{-1}\bar{A}_{vu}B(X_k) + \epsilon^{\beta-\alpha}\tilde{B}(X_k) & \bar{A}_{vv}^{-1}\bar{A}_{vu}A_{uv}(X_k) + \epsilon^{\beta-\alpha}A_{vv}(X_k) \end{pmatrix} \tag{4.21}$$

$$\tilde{b}(X_k) = \begin{pmatrix} b_u(X_k) \\ \bar{A}_{vv}^{-1}\bar{A}_{vu}b_u(X_k) + \epsilon^{\beta-\alpha}b_v(X_k) \end{pmatrix}. \tag{4.22}$$

We first establish a sequence of preliminary lemmas before we present the proof of Lemma 4.

Lemma 5. *For any $k \geq 0$, the following inequalities hold:*

$$\begin{aligned}\|\tilde{A}(X_k)\| &\leq \delta,\\ \|\bar{\tilde{A}}\| &\leq \delta,\\ \|\tilde{b}(X_k)\| &\leq \delta b_{\max},\\ \bar{\tilde{b}} &= 0,\end{aligned}$$

where $\delta = 2(1+\|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|+\epsilon^{\beta-\alpha})$, $\bar{\tilde{A}} = \lim_{k\to\infty}\tilde{A}(X_k)$, and $\bar{\tilde{b}} = \lim_{k\to\infty}\tilde{b}(X_k)$.

Proof. We begin by proving the first inequality:

$$\begin{aligned}\|\tilde{A}(X_k)\| \leq& \|B(X_k)\| + \|A_{uv}(X_k)\| + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}B(X_k)\| + \epsilon^{\beta-\alpha}\|\tilde{B}(X_k)\|\\ &+ \|\bar{A}_{vv}^{-1}\bar{A}_{vu}A_{uv}(X_k)\| + \epsilon^{\beta-\alpha}\|A_{vv}(X_k)\|\\ \leq& 1 + 1 + c + c + 2\epsilon^{\beta-\alpha}\\ =& 2(c + 1 + \epsilon^{\beta-\alpha}),\end{aligned} \tag{4.23}$$

where $c = \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|$ and the last inequality follows from the assumptions. Similarly, one can also show the remaining inequalities. □

Lemma 6. *For Θ_τ and Θ_0, the following inequalities hold:*

$$\begin{aligned}\|\Theta_\tau - \Theta_0\| &\leq 2\tilde{\epsilon}\tau\|\Theta_0\| + 2\tilde{\epsilon}\tau b_{\max},\\ \|\Theta_\tau - \Theta_0\| &\leq 4\tilde{\epsilon}\tau\|\Theta_\tau\| + 4\tilde{\epsilon}\tau b_{\max},\\ \|\Theta_\tau - \Theta_0\|^2 &\leq 32\tilde{\epsilon}^2\tau^2\|\Theta_\tau\|^2 + 32\tilde{\epsilon}^2\tau^2 b_{\max}^2,\end{aligned}$$

where $\tilde{\epsilon} = \epsilon^\alpha\delta$.

Proof. Recall that $\delta = 2\left(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\| + \frac{\epsilon^{\beta-\alpha}}{2}\right)$, therefore we have $\tilde{\epsilon} = \epsilon^\alpha\delta$. By applying Lemma 5, we obtain

$$\|\Theta_{k+1} - \Theta_k\| = \epsilon^\alpha\|\tilde{A}(X_k)\Theta_k + \tilde{b}(X_k)\| \leq \tilde{\epsilon}(\|\Theta_k\| + b_{\max}). \tag{4.24}$$

The result then follows from the steps in the proof of Lemma 3 in [63]. □

Lemma 7. *For any $k \geq 0$, the following inequality holds*

$$\left|(\Theta_{k+1} - \Theta_k)^\top P(\Theta_{k+1} - \Theta_k)\right| \leq 2\tilde{\epsilon}^2\gamma_{\max}(\|\Theta_k\|^2 + b_{\max}^2).$$

Proof. The lemma follows directly from Equation (4.24):

$$\begin{aligned}\left|(\Theta_{k+1}-\Theta_k)^\top P(\Theta_{k+1}-\Theta_k)\right| &\leq \gamma_{\max}\|\Theta_{k+1}-\Theta_k\|^2\\ &\leq \tilde{\epsilon}^2\gamma_{\max}(\|\Theta_k\|+b_{\max})^2\\ &\leq 2\tilde{\epsilon}^2\gamma_{\max}(\|\Theta_k\|^2+b_{\max}^2).\end{aligned}$$

□

Lemma 8. *For all $k \geq \tau$, the following inequality holds:*

$$\begin{aligned}&\left|\mathbb{E}\left[\Theta_k^\top P\left(\bar{\tilde{A}}\Theta_k-\frac{1}{\epsilon^\alpha}(\Theta_{k+1}-\Theta_k)\right)\middle|\,\Theta_{k-\tau},X_{k-\tau}\right]\right|\\ \leq& 10\tilde{\epsilon}\tau\gamma_{\max}(1+6\delta)(1+b_{\max})\left(\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}]+(1+b_{\max})^2\right)\\ =&\tilde{\eta}_1\tilde{\epsilon}\tau\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}]+\tilde{\eta}_2\tilde{\epsilon}\tau.\end{aligned}$$

Proof. For ease of notation, we prove the lemma for $k=\tau$, but the proof for any $k \geq \tau$ is identical. We consider

$$\begin{aligned}&\mathbb{E}\left[\Theta_\tau^\top P\left(\bar{\tilde{A}}\Theta_\tau-\frac{1}{\epsilon^\alpha}(\Theta_{\tau+1}-\Theta_\tau)\right)\middle|\,\Theta_0,X_0\right]\\ =&\mathbb{E}\left[\Theta_\tau^\top P\left(\bar{\tilde{A}}\Theta_\tau-(\tilde{A}(X_\tau)\Theta_\tau+\tilde{b}(X_\tau))\right)\middle|\,\theta_0,X_0\right]\\ =&\mathbb{E}\left[\Theta_\tau^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)\Theta_\tau\middle|\,\Theta_0,X_0\right]-\mathbb{E}\left[\Theta_\tau^\top P\tilde{b}(X_\tau)\middle|\,\Theta_0,X_0\right].\end{aligned}\tag{4.25}$$

We first consider the first term on the RHS of the above equation:

$$\begin{aligned}&\mathbb{E}\left[\Theta_\tau^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)\Theta_\tau\middle|\,\Theta_0,X_0\right]\\ =&\mathbb{E}\left[\Theta_0^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)\Theta_0\middle|\,\Theta_0,X_0\right]\\ &+\mathbb{E}\left[(\Theta_\tau-\Theta_0)^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)(\Theta_\tau-\Theta_0)\middle|\,\Theta_0,X_0\right]\\ &+\mathbb{E}\left[(\Theta_\tau-\Theta_0)^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)\Theta_0\middle|\,\Theta_0,X_0\right]\\ &+\mathbb{E}\left[\Theta_0^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)(\Theta_\tau-\Theta_0)\middle|\,\Theta_0,X_0\right].\end{aligned}\tag{4.26}$$

We will now analyze each term on the RHS above. Starting with the first

term:

$$\begin{aligned}\mathbb{E}\left[\Theta_0^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)\Theta_0\middle|\,\Theta_0,X_0\right] &= \left|\Theta_0^\top P\left(\bar{\tilde{A}}-\mathbb{E}[\tilde{A}(X_\tau)|X_0]\right)\Theta_0\right| \\ &\leq \left\|\Theta_0^\top P\right\|\left\|\left(\bar{\tilde{A}}-\mathbb{E}[\tilde{A}(X_\tau)|X_0]\right)\Theta_0\right\|, \\ &\leq \tilde{\epsilon}\gamma_{\max}\|\Theta_0\|^2\end{aligned} \tag{4.27}$$

where the final inequality follows from the assumptions on the mixing time τ and the fact that $\left\|\begin{pmatrix} 1 & 1 \\ \bar{A}_{vv}^{-1}\bar{A}_{vu} & \bar{A}_{vv}^{-1}\bar{A}_{vu}+\epsilon^{\beta-\alpha}\end{pmatrix}\right\| \leq \delta = 2(1+\|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|+\epsilon^{\beta-\alpha})$. Next, we bound the second term on the RHS of Equation (4.26):

$$\begin{aligned}&\left|\mathbb{E}\left[(\Theta_\tau-\Theta_0)^\top P\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)(\Theta_\tau-\Theta_0)\middle|\,\Theta_0,X_0\right]\right| \\ \leq&\mathbb{E}\big[\,\|(\Theta_\tau-\Theta_0)^\top P\|\|\left(\bar{\tilde{A}}-\tilde{A}(X_\tau)\right)(\Theta_\tau-\Theta_0)\|\Big|\,\Theta_0,X_0\big] \\ \leq&\gamma_{\max}\mathbb{E}\big[\,(\|\bar{\tilde{A}}\|+\|\tilde{A}(X_\tau)\|)\|\Theta_\tau-\Theta_0\|^2\Big|\,\Theta_0,X_0\big] \\ \leq&2\delta\gamma_{\max}\mathbb{E}\big[\|\Theta_\tau-\Theta_0\|^2|\Theta_0,X_0\big],\end{aligned} \tag{4.28}$$

where the last inequality follows from Lemma 5. Finally, we bound the third and fourth terms on the RHS of Equation (4.26):

$$\begin{aligned}&\left|\mathbb{E}\big[(\Theta_\tau-\Theta_0)^\top P\big(\bar{\tilde{A}}-\tilde{A}(X_\tau)\big)\Theta_0|\Theta_0,X_0\big]\right| \\ &+\left|\mathbb{E}\big[\Theta_0^\top P\big(\bar{\tilde{A}}-\tilde{A}(X_\tau)\big)(\Theta_\tau-\Theta_0)|\Theta_0,X_0\big]\right| \\ &\qquad\leq 4\delta\gamma_{\max}\|\Theta_0\|\mathbb{E}[\|\Theta_\tau-\Theta_0\||\Theta_0,X_0] \\ &\qquad\leq 8\tilde{\epsilon}\delta\tau\gamma_{\max}\|\Theta_0\|(\|\Theta_0\|+b_{\max}) \\ &\qquad\leq 8\tilde{\epsilon}\delta\tau\gamma_{\max}\|\Theta_0\|^2+8\epsilon'\delta\tau\gamma_{\max}\|\Theta_0\|b_{\max},\end{aligned} \tag{4.29}$$

where the first inequality follows from Lemma 5 and the second inequality follows from Lemma 6.

Next we consider the second term on the RHS of Equation (4.25):

$$
\begin{aligned}
&\left| - \mathbb{E}\big[\Theta_\tau^\top P\tilde{b}(X_\tau)|\Theta_0, X_0\big]\right| \\
=&\left| - \mathbb{E}\big[\Theta_0^\top P\tilde{b}(X_\tau)|\Theta_0, X_0\big] - \mathbb{E}\big[(\Theta_\tau - \Theta_0)^\top P\tilde{b}(X_\tau)|\Theta_0, X_0\big]\right| \\
\leq&\tilde{\epsilon}\gamma_{\max}\|\Theta_0\| + \gamma_{\max}b_{\max}\mathbb{E}[\|\Theta_\tau - \Theta_0\||\Theta_0, X_0] \\
\leq&\tilde{\epsilon}\gamma_{\max}\|\Theta_0\| + 2\tilde{\epsilon}\tau\gamma_{\max}b_{\max}(\|\Theta_0\| + b_{\max}),
\end{aligned}
\tag{4.30}
$$

where the final inequality follows from Lemma 6.

Now, combining (4.27) - (4.30), we get

$$
\begin{aligned}
&\left|\mathbb{E}\big[\Theta_k^\top P\left(\bar{\tilde{A}}\Theta_k - \frac{1}{\epsilon^\alpha}(\Theta_{k+1} - \Theta_k)\right)|\Theta_{k-\tau}, X_{k-\tau}\big]\right| \\
\leq&\big(\tilde{\epsilon}\gamma_{\max} + 8\tilde{\epsilon}\delta\tau\gamma_{\max}\big)\|\Theta_0\|^2 + 2\tilde{\epsilon}\tau\gamma_{\max}b_{\max}^2 \\
&+ \big(8\tilde{\epsilon}\delta\tau\gamma_{\max}b_{\max} + \tilde{\epsilon}\gamma_{\max} + 2\tilde{\epsilon}\tau\gamma_{\max}b_{\max}\big)\|\Theta_0\| \\
&+ 2\delta\gamma_{\max}\mathbb{E}[\|\Theta_\tau - \Theta_0\|^2|\Theta_0, X_0] \\
\leq&\big(2\tilde{\epsilon}\gamma_{\max} + 8\tilde{\epsilon}\delta\tau\gamma_{\max} + \tilde{\epsilon}\tau\gamma_{\max}b_{\max} + 4\tilde{\epsilon}\delta\tau\gamma_{\max}b_{\max}\big)\|\Theta_0\|^2 \\
&+ 2\tilde{\epsilon}\tau\gamma_{\max}b_{\max} + 4\tilde{\epsilon}\delta\tau\gamma_{\max}b_{\max} + \tilde{\epsilon}\gamma_{\max} + 2\tilde{\epsilon}\tau\gamma_{\max}b_{\max}^2 \\
&+ 2\delta\gamma_{\max}\mathbb{E}[\|\Theta_\tau - \Theta_0\|^2|\Theta_0, X_0] \\
\leq&\big(2\tilde{\epsilon}\tau\gamma_{\max}(1 + 4\delta)(1 + b_{\max})\big)\|\Theta_0\|^2 + \tilde{\epsilon}\tau\gamma_{\max}\big((2b_{\max} + 1)^2 + 4\delta b_{\max}\big) \\
&+ 2\delta\gamma_{\max}\mathbb{E}[\|\Theta_\tau - \Theta_0\|^2|\Theta_0, X_0] \\
\leq&\big(2\tilde{\epsilon}\tau\gamma_{\max}(1 + 4\delta)(1 + b_{\max})\big)\mathbb{E}[\|\Theta_\tau\|^2|\Theta_0, X_0] \\
&+ \tilde{\epsilon}\tau\gamma_{\max}\big((2b_{\max} + 1)^2 + 4\delta b_{\max}\big) \\
&+ \big(\gamma_{\max}(1 + 6\delta)(1 + b_{\max})\big)\mathbb{E}[\|\Theta_\tau - \Theta_0\|^2|\Theta_0, X_0] \\
\leq&\big(2\tilde{\epsilon}\tau\gamma_{\max}(1 + 4\delta)(1 + b_{\max})\big)\mathbb{E}[\|\Theta_\tau\|^2|\Theta_0, X_0] \\
&+ \tilde{\epsilon}\tau\gamma_{\max}\big((2b_{\max} + 1)^2 + 4\delta b_{\max}\big) \\
&+ \big(\gamma_{\max}(1 + 6\delta)(1 + b_{\max})\big)\big(32\tilde{\epsilon}^2\tau^2\mathbb{E}[\|\Theta_\tau\|^2|\Theta_0, X_0] + 32\tilde{\epsilon}^2\tau^2b_{\max}^2\big) \\
\leq&10\tilde{\epsilon}\tau\gamma_{\max}(1 + 6\delta)(1 + b_{\max})\mathbb{E}[\|\Theta_\tau\|^2|\Theta_0, X_0] \\
&+ 10\tilde{\epsilon}\tau\gamma_{\max}(1 + 6\delta)(1 + b_{\max})^3,
\end{aligned}
\tag{4.31}
$$

where the second inequality follows from the fact that $2\|\theta_0\| \leq 1 + \|\theta_0\|^2$ and $\tau \geq 1$, the fourth inequality follows from the triangle inequality and the penultimate inequality follows from Lemma 6. □

Next, we lower bound the minimum eigenvalue of the matrix-valued function $\Psi(\cdot)$.

Lemma 9. *Let* $\Psi(\mu) = \begin{pmatrix} \frac{\xi_2}{\xi_1+\xi_2} & -\frac{\xi_1\xi_2}{\xi_1+\xi_2} \\ -\frac{\xi_1\xi_2}{\xi_1+\xi_2} & \frac{1}{\mu}\frac{\xi_1}{\xi_1+\xi_2} - \frac{\mu\nu\xi_1}{\xi_1+\xi_2} \end{pmatrix}$ *with* $\xi_1, \xi_2, \nu > 0$ *and* $\mu \geq 0$. *Then, the following holds*

$$\lambda_{\min}(\Psi(\mu)) \geq \kappa_1 - \kappa_2\mu$$

where $\kappa_1 = \frac{\xi_2}{\xi_1+\xi_2}$ *and* κ_2 *is a constant that depends only on* ξ_1, ξ_2 *and* ν.

Proof. The minimum eigenvalue of a 2×2 matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is

$$\frac{1}{2}(a + d - \sqrt{(a-d)^2 + 4bc}),$$

so we have

$$\begin{aligned} \lambda_{\min}(\Psi(\mu)) = \frac{1}{2}\Bigg(& \frac{\xi_2}{\xi_1+\xi_2} + \frac{\xi_1}{\xi_1+\xi_2}(\frac{1}{\mu} - \nu) \\ & - \frac{\xi_1}{\xi_1+\xi_2}\sqrt{\big(\frac{\xi_2}{\xi_1} - (\frac{1}{\mu} - \nu)\big)^2 + (2\xi_2)^2}\Bigg). \end{aligned} \tag{4.32}$$

In order to obtain a lower bound on $\lambda_{\min}(\Psi(\mu))$, we first establish an upper bound on the third term on the RHS in the above equation. Defining $f(\mu) = \mu\sqrt{\big(\frac{\xi_2}{\xi_1} - (\frac{1}{\mu} - \nu)\big)^2 + (2\xi_2)^2}$, we have

$$\begin{aligned} f'(0) &= -(\nu + \frac{\xi_2}{\xi_1}) \\ f''(\mu) &= \frac{(\nu + \frac{\xi_2}{\xi_1})^2 + 4\xi_2^2}{f(\mu)^2} - \frac{f'(\mu)^2}{f(\mu)} \\ &\leq \max_{\mu \geq 0} \frac{(\nu + \frac{\xi_2}{\xi_1})^2 + 4\xi_2^2}{f(\mu)^2} - \frac{f'(\mu)^2}{f(\mu)} = 2\kappa_2 < \infty, \end{aligned} \tag{4.33}$$

which implies that

$$\begin{aligned} f(\mu) &\leq f(0) + f'(0)\mu + \kappa_2\mu^2 \\ &= 1 - (\nu + \frac{\xi_2}{\xi_1})\mu + \kappa_2\mu^2. \end{aligned} \tag{4.34}$$

Substituting the above equation into Equation (4.32) yields

$$\begin{aligned}\lambda_{\min}(\Psi(\mu)) \geq & \frac{1}{2}\Bigg(\frac{\xi_2}{\xi_1+\xi_2}+\frac{\xi_1}{\xi_1+\xi_2}(\frac{1}{\mu}-\nu) \\ & \qquad -\frac{1}{\mu}\frac{\xi_1}{\xi_1+\xi_2}\big(1-(\nu+\frac{\xi_2}{\xi_1})\mu+\kappa_2\mu^2\big)\Bigg) \\ \geq & \frac{1}{2}\big(\frac{2\xi_1 2}{\xi_1+\xi_2}-2\kappa_2\mu\big) \\ = & \kappa_1-\kappa_2\mu.\end{aligned} \tag{4.35}$$

□

We are now ready to prove Lemma 4. For any $k \geq \tau$, we have:

$$\begin{aligned}&\mathbb{E}\left[W(\Theta_{k+1})-W(\Theta_k)|\Theta_{k-\tau},X_{k-\tau}\right] \\ =&\mathbb{E}\left[2\Theta_k^\top P(\Theta_{k+1}-\Theta_k)+(\Theta_{k+1}-\Theta_k)^\top P(\Theta_{k+1}-\Theta_k)|\Theta_{k-\tau},X_{k-\tau}\right] \\ =&\mathbb{E}[2\Theta_k^\top P(\Theta_{k+1}-\Theta_k-\epsilon^\alpha\bar{A}\Theta_k) \\ &+(\Theta_{k+1}-\Theta_k)^\top P(\Theta_{k+1}-\Theta_k)|\Theta_{k-\tau},X_{k-\tau}]+2\epsilon^\alpha\mathbb{E}[\Theta_k^\top P\bar{A}\Theta_k|\Theta_{k-\tau},X_{k-\tau}].\end{aligned}$$

Using the facts that P_u and P_v are the solutions to their respective Lyapunov equations, we have

$$\mathbb{E}\left[\Theta_k^\top P\bar{A}\Theta_k|\Theta_{k-\tau},X_{k-\tau}\right] \leq -\lambda_{\min}\mathbb{E}\left[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}\right], \tag{4.36}$$

where $\lambda_{\min}$ is the smallest eigenvalue of

$$\Psi=\frac{1}{\xi_1+\xi_2}\begin{pmatrix}\xi_2 & -\xi_1\xi_2\\ -\xi_1\xi_2 & \xi_1\left(\epsilon^{-\alpha+\beta}-2\|P_v\bar{A}_{vv}^{-1}\bar{A}_{vu}\bar{A}_{uv}\|\right)\end{pmatrix}.$$

Combining the above equation, Lemma 7 and Lemma 8 with Equation (4.36), we obtain

$$\begin{aligned}&\mathbb{E}[W(\Theta_{k+1})-W(\Theta_k)|\Theta_{k-\tau},X_{k-\tau}] \\ \leq&-2\epsilon^\alpha\lambda_{\min}\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}]+\epsilon^\alpha\left(\tilde{\eta}_1\tilde{\epsilon}\tau\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}]+\tilde{\eta}_2\tilde{\epsilon}\tau\right) \\ &+2\tilde{\epsilon}^2\gamma_{\max}\left(\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}]+b_{\max}^2\right) \\ \leq&\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau},X_{k-\tau}]\left(-2\epsilon^\alpha\lambda_{\min}+\tilde{\eta}_1\epsilon^\alpha\tilde{\epsilon}\tau+2\tilde{\epsilon}^2\gamma_{\max}\right) \\ &+\epsilon^\alpha\tilde{\epsilon}\tau\left(\tilde{\eta}_2+4\left(1+\|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|+\epsilon^{\beta-\alpha}\right)\right).\end{aligned}$$

Applying the bound on $\lambda_{\min}$ in Lemma 9, we further get

$$\begin{aligned}
&\mathbb{E}[W(\Theta_{k+1}) - W(\Theta_k)|\Theta_{k-\tau}, X_{k-\tau}] \\
\leq&\mathbb{E}[\|\Theta_k\|^2|\Theta_{k-\tau}, X_{k-\tau}] \left(-\epsilon^\alpha(\kappa_1 - \kappa_2\epsilon^{\alpha-\beta}) + \tilde{\eta}_1\epsilon^\alpha\tilde{\epsilon}\tau + 2\tilde{\epsilon}^2\gamma_{\max}\right) \\
&+ \epsilon^\alpha\tilde{\epsilon}\tau\left(\tilde{\eta}_2 + 4\left(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\| + \epsilon^{\beta-\alpha}2\right)\right) \\
\leq&\mathbb{E}\left[\|\Theta_k\|^2|\Theta_{k-\tau}, X_{k-\tau}\right]\left(-\epsilon^\alpha\left(\frac{\kappa_1}{2} - \kappa_2\epsilon^{\alpha-\beta}\right)\right) \\
&+ \epsilon^\alpha\tilde{\epsilon}\tau\left(\tilde{\eta}_2 + 4\left(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\| + \epsilon^{\beta-\alpha}\right)\right) \\
\leq&\mathbb{E}\left[\|\Theta_k\|^2|\Theta_{k-\tau}, X_{k-\tau}\right]\left(-\epsilon^\alpha(\frac{\kappa_1}{2} - \kappa_2\epsilon^{\alpha-\beta})\right) \\
&+ \epsilon^{2\beta}\tau\big((3 + 2\|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|)(\tilde{\eta}_2 + 4(1 + \|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|)) + 6 + 4\|\bar{A}_{vv}^{-1}\bar{A}_{vu}\|\big) \\
=&\mathbb{E}\left[\|\Theta_k\|^2|\Theta_{k-\tau}, X_{k-\tau}\right]\left(-\epsilon^\alpha\left(\frac{\kappa_1}{2} - \kappa_2\epsilon^{\alpha-\beta}\right)\right) + \epsilon^{2\beta}\tau\eta_2 \\
\leq& -\frac{\epsilon^\alpha}{\gamma_{\max}}\left(\frac{\kappa_1}{2} - \kappa_2\epsilon^{\alpha-\beta}\right)\mathbb{E}[W(\Theta_k)] + \epsilon^{2\beta}\tau\eta_2,
\end{aligned} \tag{4.37}$$

where the second inequality follows from the assumption on ϵ, α and β, and the third inequality follows from the fact that $\epsilon < 1$ and $\alpha > \beta$.

4.5.2 The Lyapunov Function in Equation (4.7)

The rationale behind the Laypunov function is well known to control theorists, but we present it here for the interested reader.

- Setting $\epsilon = 0$ in Equation (4.6) is equivalent to studying the system of ODEs in a slow time-scale where the fast time-scale dynamics are assumed to converge instantaneously. In this case, for a fixed u, v can be written as $v_u = -\bar{A}_{vv}^{-1}\bar{A}_{vu}u$ and substituting this expression in Equation (4.5), the ODE is purely in terms of u. The first term $u^T P_u u$ in Equation (4.7) is the standard Lyapunov function used in control theory to study the stability of the resulting ODE for u.

- The second term $\left(v + \bar{A}_{vv}^{-1}\bar{A}_{vu}u\right)^\top P_v\left(v + \bar{A}_{vv}^{-1}\bar{A}_{vu}u\right)$ studies the convergence of v to v_u for a fixed u and thus, corresponds to the stability of the fast subsystem.

4.5.3 Experimental Setup Details

Following is a detailed description of the reinforcement learning problems we implemented:[1]

1. **Mountain Car:** In the basic mountain car problem, an underpowered car is positioned in a valley between two mountains on a one-dimensional track. The aim of the problem is to drive the car to the top of the mountain on the right-hand side, but the engine power available is insufficient to simply accelerate and power through to the top. Therefore, a player has to build up momentum by going back and forth between the two mountains until the car has sufficient momentum to reach its goal. The state space, action space, cost structure and initialization details for the mountain car problem are as follows:

   - *State Space:*
     (Car Position, Car Velocity) $\in [-1.2, 0.6] \times [-0.07, 0.07]$.
   - *Action Space:* 0, 1 and 2 (denoting left, no and right acceleration respectively).
   - *Cost Structure:* $+1$ cost incurred for every time step the car has not achieved its goal. 0 cost incurred upon reaching the goal.
   - *Initialization/Starting State:* The car's position is initialized to a random value in $[-0.6, 0.4]$. Its velocity is initialized to 0.

2. **Inverted Pendulum:** In the classic inverted pendulum swing-up problem, a frictionless pendulum is hinged/pivoted at one end and the aim of the problem is to keep the pendulum in an upright position (with respect to the pivot) for as long as possible by applying a torque at the pivot point (sometimes referred to as the joint effort). The state space, action space, cost structure and initialization details for the inverted pendulum problem are as follows:

   - *State Space:*
     $(\cos(\theta), \sin(\theta), \dot{\theta}) \in [-1.0, 1.0] \times [-1.0, 1.0] \times [-8.0, 8.0]$. Here, $\theta \in [-\pi, \pi]$ denotes the angular position of the pendulum with respect to the pivot.

[1] We used the OpenAI Gym implementation of these environments, available at `https://gym.openai.com/`.

- *Action Space:* Torque $\in [-2.0, 2.0]$.
- *Cost Structure:* The equation associated with the cost function is the following:

$$-(\theta^2 + 0.1\dot{\theta} + 0.001 \times \text{torque}^2).$$

- *Initialization/Starting State:* The pendulum's angular position is initialized to a random value in $[-\pi, \pi]$. Its angular velocity is initialized to a random value $\in [-1, 1]$.

4.5.4 Slope Calculations

Bounding $\mathbb{E}[|\psi_N|]$:

We have the following N points: $\{X_i = i, Y_i = \|\Theta_{k+i} - \Theta_0\|\}_{i=1}^N$. Using the formula for the slope of the best-fit line passing through these points, we get:

$$\psi_N = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^N (X_i - \bar{X})^2}, \tag{4.38}$$

where $\bar{X} = \frac{\sum_{i=1}^N X_i}{N} = \frac{1}{N}\sum_{i=1}^N X_i = \frac{N+1}{2}$ and $\bar{Y} = \frac{\sum_{i=1}^N Y_i}{N}$. Also, note that $\sum_{i=1}^N (X_i - \bar{X})^2 = \sum_{i=1}^N (i - \frac{N+1}{2})^2 = \frac{N(N-1)(N+1)}{12}$. Therefore, we have

$$\mathbb{E}[\psi_N] = \frac{12\sum_{i=1}^N (i - \frac{N+1}{2})\mathbb{E}[(Y_i - \bar{Y})]}{N(N-1)(N+1)}. \tag{4.39}$$

From Equation (4.18) we know that $d - \sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max} c}} \leq \mathbb{E}[Y_i] \leq d + \sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max} c}}$. This also implies that $d - \sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max} c}} \leq \mathbb{E}[\bar{Y}] \leq d + \sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max} c}}$. Using these

two facts in Equation (4.39)

$$
\begin{aligned}
|\mathbb{E}[\psi_N]| &\leq \frac{24\left(\sum_{i=1}^{\lfloor\frac{N+1}{2}\rfloor}(\frac{N+1}{2}-i)+\sum_{i=\lfloor\frac{N+1}{2}\rfloor+1}^{N}(i-\frac{N+1}{2})\right)\sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max}c}}}{N(N-1)(N+1)} \\
&\leq \frac{24\left(\sum_{i=1}^{\lfloor\frac{N+1}{2}\rfloor}(\frac{N+1}{2}-i)+\sum_{i=1}^{N-\lfloor\frac{N+1}{2}\rfloor}i\right)\sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max}c}}}{N(N-1)(N+1)} \\
&\leq \frac{24\frac{(N+1)^2}{4}\sqrt{\frac{2K_2\mu^{2-\lambda}}{\gamma_{\max}c}}}{N(N-1)(N+1)} = O\left(\frac{\mu^{1-\frac{\lambda}{2}}}{N}\right),
\end{aligned}
$$

where the second inequality follows from centering the second summation term in the numerator and the last inequality follows from the fact that $\sum_{i=1}^{\lfloor\frac{N+1}{2}\rfloor}-i+\sum_{i=1}^{N-\lfloor\frac{N+1}{2}\rfloor}i\leq 0$.

Bounding $\mathrm{Var}(\psi_N)$:

Using Equation (4.38):

$$
\begin{aligned}
\mathbb{E}[\psi_N^2] &= \frac{\mathbb{E}[\left(\sum_{i=1}^{N}(X_i-\bar{X})(Y_i-\bar{Y})\right)^2]}{\left(\sum_{i=1}^{N}(X_i-\bar{X})^2\right)^2} \\
&\leq \frac{\sum_{i=1}^{N}(X_i-\bar{X})^2\mathbb{E}[\sum_{i=1}^{N}(Y_i-\bar{Y})^2]}{\left(\sum_{i=1}^{N}(X_i-\bar{X})^2\right)^2} \\
&= \frac{\mathbb{E}[\sum_{i=1}^{N}(Y_i-\bar{Y})^2]}{\sum_{i=1}^{N}(X_i-\bar{X})^2} \\
&\leq \frac{24\mathbb{E}[\sum_{i=1}^{N}(Y_i^2+\bar{Y}^2)]}{N(N-1)(N+1)} \\
&\leq \frac{24\mathbb{E}[\sum_{i=1}^{N}(Y_i^2+\frac{\sum_{i=1}^{N}Y_i^2}{N})]}{N(N-1)(N+1)} \\
&\leq \frac{48\left(\frac{4K_2\mu^{2-\lambda}}{\gamma_{\max}c}+2\|\Theta_0-\Theta^*\|^2+\right)}{(N-1)(N+1)} = O(\frac{1}{N^2}),
\end{aligned}
\tag{4.40}
$$

where the first inequality follows from the Cauchy-Schwarz inequality, the second inequality follows from the fact that $(a+b)^2\leq 2a^2+2b^2$ and $\sum_{i=1}^{N}(X_i-\bar{X})^2=\sum_{i=1}^{N}(i-\frac{N+1}{2})^2=\frac{N(N-1)(N+1)}{12}$, the third inequality follows from Cauchy-Schwarz inequality and the final inequality follows from Equations

(4.16) - (4.17) and the fact that $(a+b)^2 \leq 2a^2 + 2b^2$.

CHAPTER 5

DOUBLE Q-LEARNING ANALYSIS AND INSIGHTS

5.1 Introduction

Reinforcement learning (RL) seeks to design efficient algorithms to find optimal policies for Markov Decision Processes (MDPs) without any knowledge of the underlying model (known as model-free learning) [4]. In this thesis, we study the performance of Double Q-learning [71, 72], which is a popular variant of the standard Watkins's model-free Q-learning algorithm [73, 74]. Double Q-learning was proposed to remedy the stability issues associated with the standard Q-learning algorithm (due to maximization bias of the Q-function) by using two estimators instead of one. It has been shown empirically that Double Q-learning finds a better policy in the tabular setting [71] and converges faster when coupled with deep neural networks for function approximation [72]. Several variations of Double Q-learning were proposed in [75, 76]. However, to the best of our knowledge, there has been no analysis of Double Q-learning vis-à-vis how it performs theoretically as compared to standard Q-learning. Our objective in this thesis is to address this question by providing a tight theoretical comparison between Double Q-learning and Q-learning while also drawing experimental insights that allow us to corroborate the theory.

Stochastic Approximation (SA) has proven to be a powerful framework to analyze reinforcement learning algorithms [57, 77, 56]. Several different types of guarantees for various reinforcement learning algorithms have been established using techniques from stochastic approximation. The most commonplace result is the asymptotic convergence of algorithms by analyzing the stability of an associated ODE. Examples include [78], [47] for classical TD-learning with linear function approximation, [79] for synchronous Q-learning, [80] for Double TD-learning, and [81, 82] for Q-learning with linear function

approximation. To the best of our knowledge, establishing the convergence of Double Q-learning with linear function approximation remains an open problem [80]. Although establishing asymptotic convergence of an algorithm is a useful theoretical goal, quantifying the finite-time convergence rate of an algorithm can be more useful in providing actionable insight to practitioners. There has been a significant body of recent work in this context. Finite-time analyses of TD-learning with either decaying or constant learning rate can be found in [63, 11, 83, 84, 85, 86]. Finite-time error bounds for synchronous Q-learning can be found in [87, 88] and for asynchronous Q-learning in [89]. This line of work primarily focuses on providing upper bounds on the error, thereby failing to make a tight comparison between a pair of algorithms designed for solving the same problem. Recently, several papers developed tight error bounds for SA and RL algorithms, including [90, 91, 92, 93].

In this thesis, we focus on comparing Double Q-learning with standard Q-learning, both theoretically and experimentally. We observe that through a particular linearization technique introduced in [90], both Double Q-learning and Q-learning can be formulated as instances of Linear Stochastic Approximation (LSA). We further utilize a recent result [92] that characterizes the asymptotic variance of an LSA recursion by a Lyapunov equation. By analyzing these associated Lyapunov equations for both Q-learning and Double Q-learning, we establish bounds comparing these two algorithms.

The main contributions of this work are twofold:

(1) Theoretical Contributions: We consider asynchronous Double Q-learning and Q-learning with linear function approximation with decaying step-size rules (as special cases of the more general LSA paradigm). Under the assumptions that the optimal policy is unique, both the algorithms converge and the step-size for Double Q-learning is twice that of Q-learning, we show that the asymptotic mean-squared errors of the two estimators of Double Q-learning are strictly worse than that of the estimator in Q-learning, while the asymptotic mean-squared error of the average of the Double Q-learning estimators is indeed equal to that of the Q-learning estimator. This result brings interesting practical insight, leading to our second set of contributions.

(2) Experimental Insights: Combining results from our experiments and previous work, we have the following observations:

1. If Double Q-learning and Q-learning use the same step-size rule, Q-learning has a faster rate of convergence initially but suffers from a higher mean-squared error. This phenomenon is observed both in our simulations and in earlier work on variants of Double TD-learning [80].

2. If the step-size used for Double Q-learning is twice that of Q-learning, then Double Q-learning achieves faster initial convergence rate, at the cost of a possibly worse mean-squared error than Q-learning. However, if the final output is the average of the two estimators in Double Q-learning, then its asymptotic mean-squared error is the same as that of Q-learning.

The thumb rule that these observations suggest is that one should use a higher learning rate for Double Q-learning while using the average of its two estimators as the output.

5.2 Q-Learning and Double Q-Learning

Consider a Markov Decision Process (MDP) specified by $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Here $\mathcal{S}$ is the finite state space, $\mathcal{A}$ is the finite action space, $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$ is the action-dependent probability transition matrix, $R \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ is the reward matrix, and $\gamma \in [0, 1)$ is the discount factor. Upon selecting an action a at state s, the agent will transit to the next state s' with probability $P((s, a), s')$ and receive an immediate reward $R(s, a)$.

A policy is a mapping from a state to an action, which specifies the action to be taken at each state. It is well known that the optimal policy can be obtained by solving the so-called Bellman equation [48, 4] for the state-action value function, also called the Q-function:

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P((s, a), s') \max_{a' \in \mathcal{A}} Q^*(s', a'). \tag{5.1}$$

In reinforcement learning, the goal is to estimate the Q-function from samples, without knowing the parameters of the underlying MDP. For simplicity, we assume the MDP is operated under a fixed behavioral policy, and we observe a sample trajectory of the induced Markov chain $\{(S_1, A_1), \cdots, (S_n, A_n), \cdots\}$. Let $X_n = (S_n, A_n)$ and define $\mathcal{X} = \mathcal{S} \times \mathcal{A}$. Since the state space

could be fairly large, function approximation is typically used to approximate the Q-function. In this work, we focus on linear function approximation due to its tractability. The goal is to find an optimal estimator $\theta^* \in \mathbb{R}^d$, such that $Q^* \approx \Phi^\top \theta^*$, where $\Phi = (\phi(s^1, a^1), \cdots, \phi(s^{|\mathcal{X}|}, a^{|\mathcal{X}|})) \in \mathbb{R}^{d \times |\mathcal{X}|}$, and $\phi(s, a) \in \mathbb{R}^d$ are given feature vectors associated with state-action pairs.

5.2.1 Q-Learning

We first consider asynchronous Q-learning [73, 74] with linear function approximation. Let $\Phi = (\phi(x_1), \cdots, \phi(x_{|\mathcal{X}|})) \in \mathbb{R}^{d \times |\mathcal{X}|}$ be the matrix consisting of columns of feature vectors. We let π_θ denote the greedy policy with respect to the parameter vector θ, i.e., $\pi_\theta(s) = \arg\max_a \phi(s, a)^T \theta$, where we assume that we break ties in the maximization according to some known rule. For ease of notation, we define $H(\theta_1, \theta_2, s) := \phi(s, \pi_{\theta_1}(s))^\top \theta_2$. This function estimates the Q-function based on θ_2 while the action is selected from the greedy policy given by θ_1. When observations on the sample path proceed to (X_n, S_{n+1}), Q-learning updates the parameter θ according to the equation:

$$\theta_{n+1} = \theta_n + \alpha_n \phi(X_n) \left(R(X_n) + \gamma H(\theta_n, \theta_n, S_{n+1}) - \phi(X_n)^\top \theta_n \right), \tag{5.2}$$

where α_n is an appropriately chosen step-size, also known as the learning rate.

5.2.2 Double Q-Learning

To improve the performance of Q-learning, Double Q-learning was introduced in [71, 72]. We consider the Double Q-learning algorithm with linear function approximation here. Double Q-learning maintains two estimators θ_n^A, θ_n^B, which are updated to estimate Q^* based on the sample path $\{X_n\}$ in the following manner:

$$\begin{aligned} \theta_{n+1}^A &= \theta_n^A + \beta_n \delta_n \left(\phi(X_n) \left(R(X_n) + \gamma H(\theta_n^A, \theta_n^B, S_{n+1}) - \phi(X_n)^\top \theta_n^A \right) \right) \\ \theta_{n+1}^B &= \theta_n^B + (1 - \beta_n) \delta_n \left(\phi(X_n) \left(R(X_n) + \gamma H(\theta_n^B, \theta_n^A, S_{n+1}) - \phi(X_n)^\top \theta_n^B \right) \right), \end{aligned} \tag{5.3}$$

where β_n are IID Bernoulli random variables equal to one w.p. $1/2$ and δ_n is the step-size. Note that at each time instant, only one of θ_A or θ_B is updated.

5.2.3 Linear Stochastic Approximation

Under the assumptions that the optimal policy is unique, the ordinary differential equation (ODE) associated with Q-learning is stable and other technical assumptions, it has been argued in [90] that the asymptotic variance of Q-learning can be studied by considering the recursion

$$\theta_{n+1} = \theta_n + \alpha_n \phi(X_n) \left(R(X_n) + \gamma \phi(S_{n+1}, \pi^*(S_{n+1}))^\top \theta_n - \phi(X_n)^\top \theta_n \right), \quad (5.4)$$

where π^* is the optimal policy π_{θ^*} based on θ^*. Here and throughout, as in [90], we assume that the Q-learning and Double Q-learning algorithms converge to some θ^*. We refer the reader to [90] for details.

Using a similar argument, one can show that the asymptotic variance of Double Q-learning can be studied by considering the following recursions:

$$\begin{aligned} \theta^A_{n+1} &= \theta^A_n + \beta_n \delta_n \left(\phi(X_n) \left(R(X_n) + \gamma \phi(S_{n+1}, \pi^*(S_{n+1}))^\top \theta^B_n - \phi(X_n)^\top \theta^A_n \right) \right) \\ \theta^B_{n+1} &= \theta^B_n + (1-\beta_n) \delta_n \big(\phi(X_n) \big(R(X_n) + \gamma \phi(S_{n+1}, \pi^*(S_{n+1}))^\top \theta^A_n \\ &\qquad - \phi(X_n)^\top \theta^B_n \big) \big). \end{aligned} \quad (5.5)$$

Our comparison of the asymptotic mean-squared errors of Q-learning and Double Q-learning will use Equations (5.4)-(5.5). In practice, however, one is typically interested in how quickly one learns the optimal policy which cannot be measured very well by using mean-squared error as the metric. But, we will see later that our simulations indicate that the insights we obtain from mean-squared error analysis hold even for learning the optimal policy.

5.3 Main Results

In this section, we present our main results. We will first review the results on asymptotic variance of linear stochastic approximation in [92] and then use these to compare the asymptotic variances of Q-learning and Double Q-learning.

5.3.1 Preliminaries

Consider the linear stochastic approximation recursion:

$$\xi_{n+1} = \xi_n + \frac{g}{n} \left(A(Y_n)\xi_n + b(Y_n) \right), \tag{5.6}$$

where g is a positive constant, Y_n is an irreducible, aperiodic Markov chain on a finite state space, A and b are a random matrix and a random vector, respectively, which are determined by Y_n. Without loss of generality, we assume ξ_n converges to $\xi^* = 0$. If $\xi^* \neq 0$, we can subtract ξ^* from ξ_n. Define the asymptotic covariance of ξ_n to be

$$\Sigma_\infty = \lim_{n\to\infty} n\mathbb{E}\left[\xi_n \xi_n^T\right].$$

The following result is from [92].

Theorem 6. *Suppose that $\bar{A} := \mathbb{E}\left[A(Y_\infty)\right]$, and $\frac{1}{2}I + g\bar{A}$ is a Hurwitz matrix, i.e., its eigenvalues have negative real parts, and $\Sigma_b := \sum_{n=2}^{\infty} \mathbb{E}\left[b(Y_n)b(Y_1)^\top\right]$, where Y_∞ is notation for a random variable with the same distribution as the stationary distribution of the Markov chain $\{Y_n\}$. Then, Σ_∞ is the unique solution to the Lyapunov equation*

$$\Sigma_\infty \left(\frac{1}{2}I + g\bar{A}^\top \right) + \left(\frac{1}{2}I + g\bar{A} \right) \Sigma_\infty + g^2 \Sigma_b = 0. \tag{5.7}$$

In the next subsection, we use the above result to establish the relationship between the asymptotic covariances of Q-learning and Double Q-learning.

5.3.2 Comparison of Q-Learning and Double Q-Learning

Throughout this section, we assume that $\theta^* = 0$ without loss of generality. If $\theta^* \neq 0$, the results can hold by subtracting θ^* from the estimators of Q-learning and Double Q-learning. Our main result is stated in the following theorem.

Theorem 7. *Define the asymptotic mean-squared error of Q-learning to be*

$$\mathrm{AMSE}(\theta) := \lim_{n\to\infty} n\mathbb{E}\left[\theta_n^T \theta_n\right],$$

the asymptotic mean-squared error of the first (or second) estimator in Double Q-learning to be

$$\text{AMSE}(\theta^A) := \lim_{n\to\infty} n\mathbb{E}\left[(\theta_n^A)^\top \theta_n^A\right],$$

and the asymptotic mean-squared error of the average of the two Double Q-learning estimators to be

$$\text{AMSE}\left(\frac{\theta^A + \theta^B}{2}\right) = \lim_{n\to\infty} \frac{1}{4} n\mathbb{E}\left[(\theta_n^A + \theta_n^B)^\top (\theta_n^A + \theta_n^B)\right].$$

Let the step-sizes of Q-learning and Double Q-learning be $\alpha_n = g/n$ and $\delta_n = 2g/n$, where g is a positive constant. Then there exists some $g_0 > 0$, such that for any $g > g_0$, the following results hold:

1. $\text{AMSE}(\theta^A) \geq \text{AMSE}(\theta)$, *and*
2. $\text{AMSE}(\frac{\theta^A+\theta^B}{2}) = \text{AMSE}(\theta)$.

Before we present the proof of the above result, we make some remarks.

Remark 3. *The condition $g > g_0$ is tied to the sufficient conditions for the stability of the ODEs associated with the covariance equations of Q-learning and Double Q-learning [92]. For instance, in the tabular case, i.e., when Φ is an identity matrix with dimension $|\mathcal{X}|$, the results hold as long as $g > \frac{1}{\mu_{\min}(1-\gamma)}$, where $\mu_{\min}$ is the minimum steady-state probability of any state $x \in \mathcal{X}$ for the stationary distribution μ. $g > \frac{1}{\mu_{\min}(1-\gamma)}$ is a common assumption used in the analysis of tabular Q-learning [89].*

Remark 4. *As mentioned in the introduction to this chapter, Double Q-learning can be slower initially due to the fact that only half the samples are used to estimate each of its estimators. One way to speed up the initial convergence rate is to double the learning rate. Our results here show that the asymptotic mean-squared error of Double Q-learning in that case will be at least as large as that of Q-learning; however, if the output of Double Q-learning is the average of its two estimators, the asymptotic mean-squared error is exactly equal to that of Q-learning with half the learning rate. Thus, Double Q-learning learns faster without sacrificing asymptotic mean-squared error. This suggests that increasing the learning rate of Double Q-learning while averaging the output can have significant benefits, which we verify using*

simulations in the next section. Now, we are ready to present the proof of the theorem.

Proof of Theorem 7: Recall from Section 5.2.3 that the asymptotic variance of Q-learning can be studied by considering the following recursion:

$$\theta_{n+1} = \theta_n + \alpha_n \phi(X_n) \left(R(X_n) + \gamma \phi(S_{n+1}, \pi^*(S_{n+1}))^\top \theta_n - \phi(X_n)^\top \theta_n \right). \quad (5.8)$$

Similarly, one can show that the asymptotic variance of Double Q-learning can be studied by considering the following recursions:

$$\begin{aligned} \theta^A_{n+1} &= \theta^A_n + \beta_n \delta_n \left(\phi(X_n) \left(R(X_n) + \gamma \phi(S_{n+1}, \pi^*(S_{n+1}))^\top \theta^B_n - \phi(X_n)^\top \theta^A_n \right) \right), \\ \theta^B_{n+1} &= \theta^B_n + (1-\beta_n) \delta_n \big(\phi(X_n) \big(R(X_n) + \gamma \phi(S_{n+1}, \pi^*(S_{n+1}))^\top \theta^A_n \\ &\qquad - \phi(X_n)^\top \theta^B_n \big) \big). \end{aligned} \quad (5.9)$$

For ease of notation, let $Z_n = (X_n, S_{n+1})$. It is shown in [87] that $\{Z_n\}$ is also an aperiodic and irreducible Markov chain. Let us define the following:

$$\begin{aligned} b(Z_n) &= \phi(X_n) R(X_n), \\ A_1(Z_n) &= \phi(X_n) \phi(X_n)^\top, \\ A_2(Z_n) &= \gamma \phi(X_n) \phi(S_{n+1}, \pi^*(S_{n+1}))^\top, \\ A(Z_n) &= A_2(Z_n) - A_1(Z_n). \end{aligned}$$

Using these definitions, we can rewrite Equations (5.8) and (5.9) as:

$$\theta_{n+1} = \theta_n + \alpha_n \left(b(Z_n) + A_2(Z_n)\theta_n - A_1(Z_n)\theta_n \right), \quad (5.10)$$

and

$$\begin{aligned} \theta^A_{n+1} &= \theta^A_n + \beta_n \delta_n \left(b(Z_n) + A_2(Z_n)\theta^B_n - A_1(Z_n)\theta^A_n \right), \\ \theta^B_{n+1} &= \theta^B_n + (1-\beta_n) \delta_n \left(b(Z_n) + A_2(Z_n)\theta^A_n - A_1(Z_n)\theta^B_n \right), \end{aligned} \quad (5.11)$$

respectively. Let $U_n = ((\theta_n^A)^\top, (\theta_n^B)^\top)^\top$. We can further rewrite Equation (5.11) in a more compact form as:

$$U_{n+1} = U_n + \alpha_n \Bigg[\begin{pmatrix} -2\beta_n A_1(Z_n) & 2\beta_n A_2(Z_n) \\ 2(1-\beta_n)A_2(Z_n) & -2(1-\beta_n)A_1(Z_n) \end{pmatrix} U_n + \begin{pmatrix} 2\beta_n b(Z_n) \\ 2(1-\beta_n)b(Z_n) \end{pmatrix} \Bigg]. \tag{5.12}$$

Let μ denote the steady-state probability vector for the Markov chain $\{X_n\}$. Let D be a diagonal matrix of dimension $|\mathcal{X}|$ such that $D_{ii} = \mu_i$. We have $\bar{A}_1 = \mathbb{E}[A_1(Z_\infty)] = \Phi D \Phi^\top, \bar{A}_2 = \mathbb{E}[A_2(Z_\infty)] = \gamma \Phi D P S_{\pi^*} \Phi^\top$, where S_{π^*} is the action selection matrix of the optimal policy π^* such that $S_{\pi^*}(s,(s,\pi^*(s))) = 1$ for $s \in \mathcal{S}$. Denote $\bar{A} = \bar{A}_2 - \bar{A}_1$.

We will now use Theorem 6 to prove our result. Let $\Sigma_\infty^Q = \lim_{n\to\infty} n\mathbb{E}\left[\theta_n \theta_n^T\right]$ and $\Sigma_\infty^D = \lim_{n\to\infty} n\mathbb{E}\left[U_n U_n^T\right]$. Clearly, $\text{AMSE}(\theta) = \text{Tr}(\Sigma_\infty^Q)$. Applying Theorem 6 to Equations (5.10) and (5.12):

$$\Sigma_\infty^Q \left(\frac{1}{2}I + g\bar{A}^\top\right) + \left(\frac{1}{2}I + g\bar{A}\right)\Sigma_\infty^Q + g^2(B_1 + B_2) = 0, \tag{5.13}$$

and

$$\Sigma_\infty^D \left(\frac{1}{2}I + g\bar{A}_D^\top\right) + \left(\frac{1}{2}I + g\bar{A}_D\right)\Sigma_\infty^D + g^2\Sigma_b^D = 0, \tag{5.14}$$

where $B_1 = \mathbb{E}\left[\sum_{n=1}^\infty b(X_n)b(X_1)^\top\right]$, $B_2 = \mathbb{E}\left[\sum_{n=2}^\infty b(X_n)b(X_1)^\top\right]$, $\bar{A}_D = \begin{pmatrix} -\bar{A}_1 & \bar{A}_2 \\ \bar{A}_2 & -\bar{A}_1 \end{pmatrix}$, and $\Sigma_b^D = 2\begin{pmatrix} B_1 & B_2 \\ B_2 & B_1 \end{pmatrix}$. Because of the symmetry in the two estimators comprising Double Q-learning, we observe that Σ_∞^D will have the following structure: $\Sigma_\infty^D = \begin{pmatrix} V & C \\ C & V \end{pmatrix}$, where

$$V = \lim_{n\to\infty} n\mathbb{E}\left[\theta_n^A(\theta_n^A)^\top\right] = \lim_{n\to\infty} n\mathbb{E}\left[\theta_n^B(\theta_n^B)^\top\right], \qquad C = \lim_{n\to\infty} n\mathbb{E}\left[\theta_n^A(\theta_n^B)^\top\right].$$

Coupling this observation with Equation (5.14) yields

$$\begin{pmatrix} V & C \\ C & V \end{pmatrix} + g\begin{pmatrix} V & C \\ C & V \end{pmatrix}\begin{pmatrix} -\bar{A}_1 & \bar{A}_2 \\ \bar{A}_2 & -\bar{A}_1 \end{pmatrix}^T + g\begin{pmatrix} -\bar{A}_1 & \bar{A}_2 \\ \bar{A}_2 & -\bar{A}_1 \end{pmatrix}\begin{pmatrix} V & C \\ C & V \end{pmatrix} + 2g^2\begin{pmatrix} B_1 & B_2 \\ B_2 & B_1 \end{pmatrix} = 0. \tag{5.15}$$

Summing the first two blocks (row-wise) of matrices in the above equation, we get

$$V+C+g(V+C)(\bar{A}_2-\bar{A}_1)^T+g(\bar{A}_2-\bar{A}_1)(V+C)+2g^2(B_1+B_2)=0. \quad (5.16)$$

Next, define $g_0 := \inf\{g \geq 0 : g\max(\lambda_{\max}(\bar{A}), \lambda_{\max}(\bar{A}_D)) < -1\}$, where $\lambda_{\max}(A)$ denotes the real part of the maximum eigenvalue of A. Note that g_0 exists since both $\bar{A}$ and $\bar{A}_D$ are Hurwitz, under the assumption that Q-learning and Double Q-learning both converge [94]. As a result, for any $g > g_0$, $\frac{1}{2}I + g\bar{A}$ is Hurwitz. Therefore, the solution $V + C$ to the above equation and the solution Σ_∞ in Equation (5.13) are unique [94]. Similarly, we also note that the solution in Equation (5.15) is also unique as $\frac{1}{2}I + g\bar{A}_D$ is Hurwitz whenever $g > g_0$.

Comparing the above equation with Equation (5.13), we get $\Sigma^Q_\infty = \frac{V+C}{2}$. Next, we observe that $\mathrm{Tr}(V) \geq \mathrm{Tr}(C)$. The reasoning behind that is as follows:

$$\lim_{n\to\infty} n\mathbb{E}\left[(\theta^A_n-\theta^B_n)^T(\theta^A_n-\theta^B_n)\right] \geq 0$$
$$\Rightarrow 2\lim_{n\to\infty} n\big\{\mathbb{E}\left[(\theta^A_n)^T\theta^A_n\right]-\mathbb{E}\left[(\theta^B_n)^T\theta^A_n\right]\big\} \geq 0 \Rightarrow \mathrm{Tr}(V)-\mathrm{Tr}(C) \geq 0,$$

where the second inequality follows from the symmetry in the two estimators comprising Double Q-learning. Using $\mathrm{Tr}(V) \geq \mathrm{Tr}(C)$, we get $\mathrm{Tr}(V) \geq \mathrm{Tr}\left(\frac{V+C}{2}\right) = \mathrm{Tr}(\Sigma^Q_\infty)$. This equation proves our first result. To prove the second result, we observe that

$$\mathrm{AMSE}\left(\frac{\theta^A+\theta^B}{2}\right) = \frac{1}{2}\mathrm{AMSE}(\theta^A)+\frac{1}{2}\mathrm{Tr}(C) = \frac{1}{2}(\mathrm{Tr}(V)+\mathrm{Tr}(C)) = \mathrm{Tr}(\Sigma^Q_\infty).$$

□

5.4 Numerical Results

In this section, we provide numerical comparisons between Double Q-learning and Q-learning on Baird's example [95], GridWorld [96], CartPole [97] and

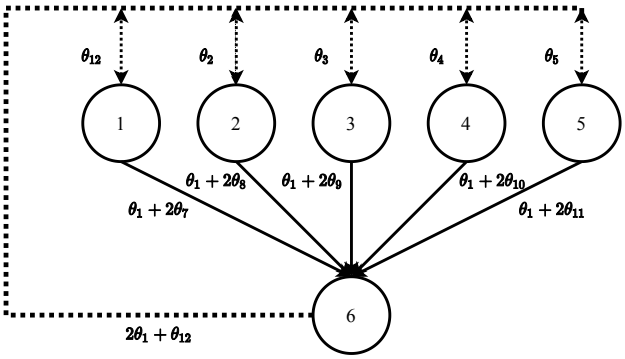
an example of maximization bias from [4]. [1] We investigate four algorithms: (1) Q-learning using step-size α_n, denoted as Q in plots; (2) Double Q-learning using step-size α_n, denoted as D-Q; (3) Double Q-learning using step-size equal to $2\alpha_n$, denoted as D-Q with twice the step-size; (4) Double Q-learning using step-size equal to $2\alpha_n$ and returning the average estimator $(\theta_n^A + \theta_n^B)/2$, denoted as D-Q avg with twice the step-size. For the vanilla Double Q-learning, we always use θ_n^A as the output estimator.

For the first two experiments, we plot the logarithm of the mean-squared error for each algorithm. We set the step-size $\alpha_n = \frac{1000}{n+10000}$. The optimal estimator θ^* is calculated by solving the projected Bellman equation [82] based on the Markov chain. Sample paths start in state 1 in Baird's example, and state $(1, 1)$ in GridWorld. We use the uniformly random policy as the behavioral policy, i.e., each valid action is taken with equal probability in any given state. $\theta_1, \theta_1^A, \theta_1^B$ are initialized to the same value which is sampled uniformly at random from $[0, 2]^d$, where d is the dimension of features. Results in each plot reflect the average over 100 sample paths.

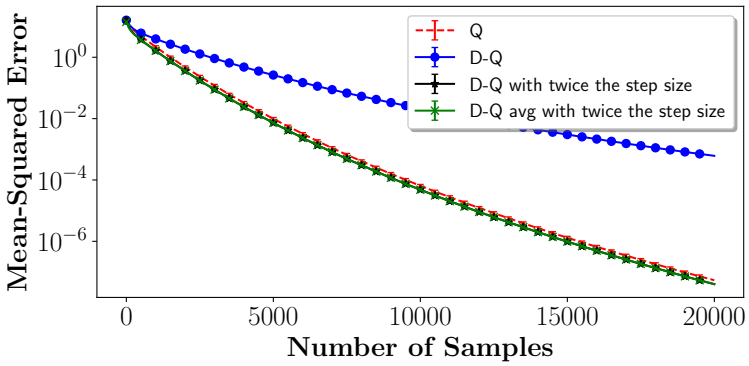
5.4.1 Baird's Example

The first environment we consider is the popular Baird's example which was used to prove that Q-learning with linear function approximation may diverge [87, 95]. It is a simple Markov chain as shown in Fig. 5.1a with six states and two actions (represented by the dotted line and the solid line respectively). When the action represented by the dotted line is taken, the agent transits to one of the first five states randomly. When an action represented by a solid line is taken, the agent transits to state 6. The Q-function is approximated by a parameter $\theta \in \mathbb{R}^{12}$, where the specific linear combination is shown next to the corresponding action in Fig. 5.1a. For the reward function $R(s, a)$, $1 \leq s \leq 6, 1 \leq a \leq 2$, we explore three different settings: (1) **Zero Reward:** the reward $R(s, a)$ is uniformly zero; (2) **Small Random Reward:** the reward $R(s, a)$ is sampled uniformly at random from $[-0.05, 0.05]$; (3) **Large Random Reward:** the reward $R(s, a)$ is sampled uniformly at random from $[-50, 50]$. Our theory applies to the Small Random Reward case and the Large Random Reward case because the optimal

[1] Code is at:
`https://github.com/wentaoweng/The-Mean-Squared-Error-of-Double-Q-Learning`



(a) Baird's Example [95]

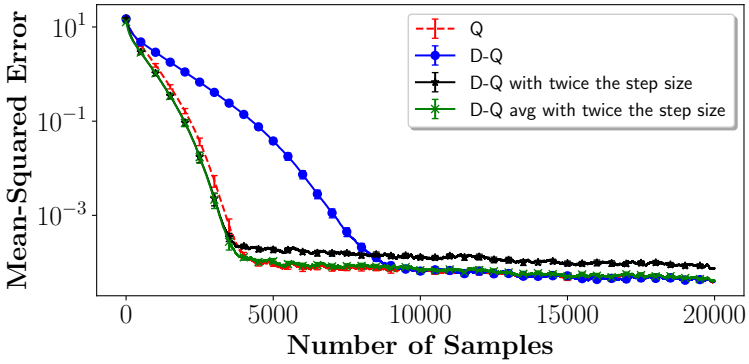


(b) Zero Reward

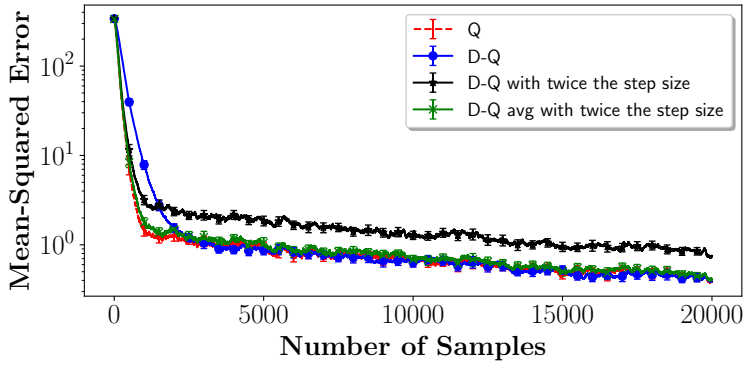
Figure 5.1: Simulation results for Baird's example. The y-axis is in logarithmic scale.

policy is unique in these two cases, but simulations indicate that our insight hold more generally even in the case of Zero Reward. Although Baird's example was originally proposed to make Q-learning diverge when γ is large, we study the case $\gamma = 0.8$ where all algorithms converge. Results are presented in Figs. 5.1b, 5.2a, and 5.2b.

In all the three scenarios, we observe that Double Q-learning converges much slower than Q-learning at an early stage, when using the same step-size. When using twice the step-size as compared to Q-learning, we observe that Double Q-learning converges slightly faster than Q-learning in Fig. 5.1b, Fig. 5.2a, and almost at the same speed in Fig. 5.2b. However, the asymptotic mean-squared error for Double Q-learning even with twice the step-size can be much worse than that of Q-learning, as shown in Fig. 5.2a and Fig. 5.2b. Finally, by simply using the averaged estimator, Double Q-learning with twice the step-size obtains both faster convergence and smaller asymptotic mean-squared error, corroborating our theory.



(a) Small Random Reward

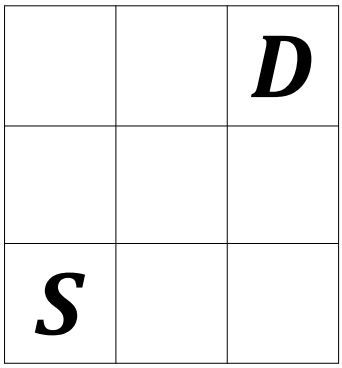


(b) Large Random Reward

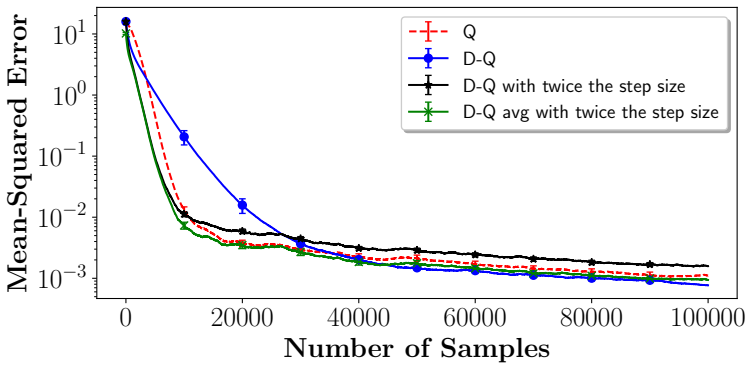
Figure 5.2: Simulation results for Baird's example. The y-axis is in logarithmic scale.

5.4.2 GridWorld

The second environment we simulate is the GridWorld game with a similar setting as in [96]. Consider a $n \times n$ grid where the agent starts at position $(1, 1)$ and the goal is to reach the position (n, n). A 3×3 GridWorld is shown in Fig. 5.3a. For each state, the agent can walk in four directions: up, down, left or right. If the agent walks out of the grid, the agent will stay at the same cell. There is a 30% probability that the chosen direction is substituted by any one of the four directions randomly. The agent receives reward -10^{-3} in each step, but receives reward 1 at the destination. The game ends when the agent arrives at the destination. We consider GridWorld with $n = 3, 4$ and 5, so the number of state-action pairs can be up to 100. The discount factor is set as $\gamma = 0.9$. We run tabular Q-learning and tabular Double Q-learning. Simulation results are shown in Fig. 5.3 and Fig. 5.4.

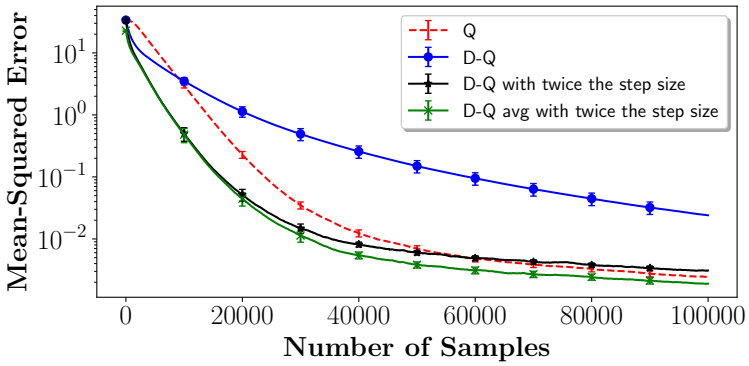


(a) An Example of 3×3 GridWorld

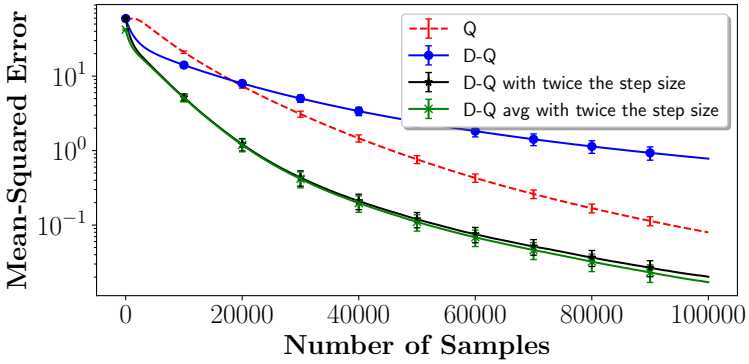


(b) 3×3 GridWorld

Figure 5.3: Simulation results for GridWorld with dimension 3. Clearly, Double Q-learning with twice the step-size and averaged output outperforms Q-learning.



(a) 4×4 GridWorld



(b) 5×5 GridWorld

Figure 5.4: Simulation results for GridWorld with dimensions 4 and 5. In both the simulations, Double Q-learning with twice the step-size and averaged output outperforms Q-learning.

As we can see from Fig. 5.3b, Double Q-learning using the same step-size as Q-learning converges much slower than all the other three algorithms even though it has a slightly better asymptotic variance. By simply doubling the step-size and using the averaged output, Double Q-learning outperforms Q-learning in all the three settings. It is worth pointing out that theoretically speaking, Theorem 7 does not apply to this example since the optimal policy in this setting is not unique. However, the insights offered by Theorem 7 still hold.

5.4.3 CartPole

The third experiment we conduct is the classical CartPole control problem introduced in [97]. In this problem, a cart with a pole is controlled by

applying a force, either to the left or to the right. The goal is to keep the pole upright for as long as possible. The player receives a +1 reward for every time step until the episode ends, which happens when the pole falls down or the cart moves out of a certain region. Unlike the previous numerical results which mainly focus on the mean-squared error, in this case, we study how fast the four algorithms can find a policy that achieves the best performance. We train the different algorithms on the CartPole-v0 environment available in OpenAI Gym [98]. Specifically, we consider Q-learning and Double Q-learning with ϵ-greedy exploration. The training is episodic, i.e., the step-size and ϵ are updated after one episode. In particular, for the nth episode, we use $\epsilon_n = \max(0.1, \min(1, 1 - \log(\frac{n}{200})))$ and $\alpha_n = \frac{40}{n+100}$. The step-size is different from previous experiments because we only train 1000 episodes for CartPole, and therefore, the step-size would have remained too large throughout if we had used the previous step-size rule leading to convergence issues. The discount factor is set as $\gamma = 0.999$. Since the state space of CartPole is continuous, we discretize it into 72 states following [99].

We evaluate the algorithms based on their "hit time", i.e., the time at which they first learn a fairly good policy. We say an algorithm has learned a fairly good policy after n steps if the mean reward of the greedy policy based on the estimator after n steps of the algorithm exceeds 195. To reduce the computational overhead, we evaluate the policy obtained after every 50 episodes by averaging the reward obtained by the policy over 1000 independently-run episodes. The distribution of the "hit time" for each algorithm in 100 independent tests is shown in Fig. 5.5. We observe that Double Q-learning using the same step-size as Q-learning performs much worse than the other algorithms. However, when using twice the step-size, Double Q-learning finds a good policy faster than Q-learning, at the cost of a larger standard deviation for the "hit time". The increase in variance can be mitigated by using the averaged estimator, which also improves the convergence rate.

5.4.4 Maximization Bias of Q-Learning

The fourth example we investigate is the maximization bias example similar to that in [4, page 135]. Since Double Q-learning was proposed to alleviate the

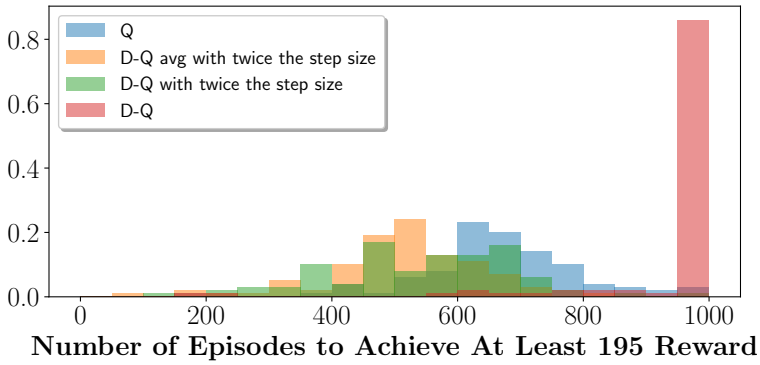


Figure 5.5: Distribution of "hit time", i.e., number of episodes needed to obtain a mean reward of 195 in CartPole-v0, with the number of episodes capped at 1000.

Table 5.1: Mean hit time along with the standard deviation for different algorithms.

Algorithm	Mean Hit Time
Q	645.0 ± 12.93
D-Q avg with twice the step-size	487.5 ± 12.19
D-Q with twice the step-size	518.0 ± 14.77

maximization bias from Q-learning. we study how the proposed modification, doubling the step-size and averaging the two estimators in Double Q-learning, affects the performance in an example where Double Q-learning is known to be helpful. To be specific, there are $M+1$ states labeled as $\{0, \cdots, M\}$ with two actions, left and right. The agent starts at state 0. If the agent goes to the right, the game ends, but if the agent moves to the left, the agent transitions to one of the other M states with equal probability. Both the actions result in zero immediate reward. When the agent is at state 1 to state M, if the action taken is to go to the right, the agent returns to state 0; if the action taken is to go to the left, the game ends. Both the actions result in a reward independently sampled from a normal distribution with mean -0.1 and standard deviation 1.

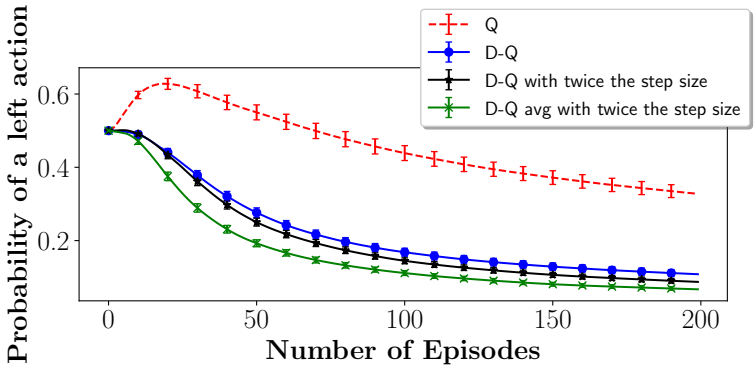
We first test the algorithms in a tabular setting with $M = 8$. The exploration policy is set to be ϵ-greedy with $\epsilon = 0.1$. In the nth episode, we let $\alpha_n = \frac{10}{n+100}$. We train the algorithms for 200 episodes. All the estimators are initialized to zero. To evaluate the algorithms, we plot the probability of the agent going left after every episode. In particular, at the end of n episodes, we track how often the estimated Q-function of the left action is larger than

that of the right action at state 0. In addition, this probability is estimated by taking the average of 1000 independent runs. Notice that going right always maximizes the mean reward for the agent, so a larger probability to go left indicates that the algorithm has learned a bad policy. The results are shown in Fig. 5.6a. As we can see, Q-learning suffers from maximization bias when the number of episodes is small since there is a large probability of going to the left. On the other hand, there is no such problem with Double Q-learning. Additionally, Double Q-learning with twice the step-size and averaging further improves the performance.

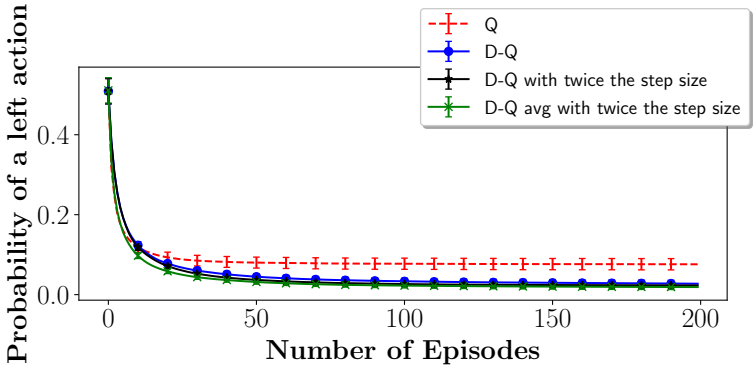
In addition to the tabular setting, we also explore a setting where neural networks are used for function approximation. In particular, we consider the same environment as before, but with $M = 10^9$. With such a large value of M, it is infeasible to maintain a table of the $Q-$function values for all state-action pairs. Instead, we assume that the $Q-$function is approximated by a neural network with two hidden layers with dimensions 4 and 8 respectively. Each pair of adjacent layers is fully connected, with ReLU as the activation function. We use stochastic gradient descent with no momentum as the optimizer. Other settings are the same as those in the tabular setting. The results are shown in Fig. 5.6b. We can see that although Q-learning does not seem to suffer from maximization bias any more, it still performs worse than Double Q-Learning. In addition, Double Q-Learning with twice the step-size and averaging helps improve the performance.

5.5 Linearization Results

In this section, we provide more details on the derivation of the results pertaining to the asymptotic mean-squared errors in Theorem 7. While [90] provides an outline of the result, we provide some missing details here, including additional assumptions under which the result in [90] is valid. We first discuss a result from [77] which will be useful to us.



(a) In a tabular setting with $M = 8$



(b) In a setting with neural network function approximation and $M = 10^9$

Figure 5.6: The probability to take the left action in state 0 for different algorithms in an environment similar to the maximization bias example from [4]. A lower probability of taking the left action indicates a better policy.

5.5.1 Central Limit Theorem for SA

Statements in this part are adapted from [77, Chapters 2 and 3]. Consider a stochastic approximation algorithm of the following form

$$\xi_n = \xi_{n-1} + \gamma_n W(\xi_{n-1}, Y_n), \tag{5.17}$$

where ξ_n lies in $\mathbb{R}^d$, and the state Y_n lies in $\mathbb{R}^k$. Suppose the algorithm satisfies following assumptions.

Assumption 1. *[77, page 43, Assumption A]*

(a) Decreasing Step-Size:

$$\gamma_n \geq 0; \ \sum_n \gamma_n = +\infty; \ \sum_n \gamma_n^\alpha < \infty \ \textit{for some } \alpha > 1. \tag{5.18}$$

(b) Markovian Noise:

There exists a Markov chain $\{\eta_n\}$, independent of $\{\xi_n\}$ with a unique stationary distribution such that $Y_n = f(\eta_n)$.

(c) Existence of a Mean Vector Field:

We assume the existence of the mean vector field defined by

$$w(\xi) := \lim_{n\to\infty} \mathbb{E}\left[W(\xi, Y_n)\right],$$

where the expectation is taken under the distribution of (Y_n).

Assumption 1(c) allows us to introduce the ODE

$$\dot{\xi} = w(\xi), \xi(0) = z \tag{5.19}$$

whose unique solution is denoted as $[\xi(z,t)]_{t\geq 0}$. The next assumption we have is on the ODE.

Assumption 2. *[77, Assumption (A.2), Assumption (A.2b)] The ODE (5.19) has an attractor ξ^*, whose domain of attraction is denoted by D_*. Assumption 1 is satisfied in D_*.*

Further, we assume the uniqueness of the attractor.

Assumption 3. *[77, page 108] The ODE is globally asymptotically stable with a unique stable equilibrium point ξ^*.*

Define

$$C(\xi) := \sum_{n=-\infty}^{+\infty} \mathrm{Cov}[W(\xi, Y_n), W(\xi, Y_1)], \tag{5.20}$$

where Cov denotes the covariance when Y_1 is stationary. We can now state the central limit theorem.

Theorem 8. *[77, page 110, Theorem 3] Suppose Assumption 2 and Assumption 3 hold, and the step-size sequence satisfies $\gamma_n = \frac{1}{n}$. If $\nabla_\xi w(\xi^*)$ and $C(\xi^*)$ exist, and $\lambda_{\max}(\nabla_\xi w(\xi^*)) < -\frac{1}{2}$, we have*

$$n^{\frac{1}{2}}(\xi_n - \xi^*) \underset{d}{\to} \mathcal{N}(0, P), \tag{5.21}$$

where P is the unique symmetric solution of the Lyapunov equation

$$\left(\frac{I}{2} + \nabla_\xi w(\xi^*)\right) P + P\left(\frac{I}{2} + \nabla_\xi w(\xi^*)^\top\right) + C(\xi^*) = 0.$$

5.5.2 Application to Q-Learning and Double Q-Learning

In this section, we show that Theorem 8 is applicable to Q-learning (defined in Equation (5.2)) and Double Q-learning (defined in Equation (5.3)) under the assumptions stated in Section 5.3. Note that the step-sizes are assumed to be $\alpha_n = \frac{g}{n}$, and $\delta_n = \frac{2g}{n}$ in Theorem 7, which are different from that in Theorem 8. Therefore, we scale the reward function and feature vectors to absorb the constant g (or $2g$) in the updates of Q-learning and Double Q-learning. The step-sizes are then shifted to $\frac{1}{n}$.

Recall $Z_n = (X_n, S_{n+1})$ defined in the proof of Theorem 7. We first notice that Assumption 1 is automatically satisfied because: (1) The step-size condition is fulfilled for $\frac{1}{n}$; (2) The samples $\{Z_n, n \geq 0\}$ form a Markov chain independent of θ_n; (3) The mean vector field $w(\theta)$ is well-defined since $\{Z_n\}$ has a unique limiting stationary distribution, and its state space $\mathcal{X} \times \mathcal{S}$ is finite. As a result, the ODE for Q-learning is defined as

$$\dot{\theta}(t) = g\mathbb{E}\left[\phi(X_n)(R(X_n) + \gamma H(\theta(t), \theta(t), S_{n+1}) - \phi(X_n)^\top \theta(t))\right], \quad (5.22)$$

and that of Double Q-learning is given by

$$\dot{\theta}^A(t) = g\mathbb{E}\left[\phi(X_n)(R(X_n) + \gamma H(\theta^A(t), \theta^B(t), S_{n+1}) - \phi(X_n)^\top \theta^A(t))\right], \quad (5.23a)$$

$$\dot{\theta}^B(t) = g\mathbb{E}\left[\phi(X_n)(R(X_n) + \gamma H(\theta^B(t), \theta^A(t), S_{n+1}) - \phi(X_n)^\top \theta^B(t))\right]. \quad (5.23b)$$

For ease of notation, denote $U(t) = ((\theta^A(t)); (\theta^B(t)))$. The notation $(\mathbf{a}; \mathbf{b})$ is a vector that is the concatenation of $\mathbf{a}$ and $\mathbf{b}$. Also, denote the right-hand side of Equation (5.22) by $w(\theta(t))$, and that of Equation (5.23) by $\tilde{w}(U(t))$.

To guarantee Assumption 2 and Assumption 3, we make the following assumption.

Assumption 4. *Both $\theta(t)$ and $U(t)$ have unique globally asymptotically stable (GAS) equilibrium points.*

Sufficient conditions under which Q-learning with linear function approximation satisfies Assumption 4 are studied in [82, 81]. While little is known on the convergence of Double Q-learning with linear function approximation, it is commonly perceived that Double Q-learning is more stable than Q-learning even when equipped with neural networks [72].

Denote the unique stable point of $\theta(t)$ as θ^*, and that of $U(t)$ as U^*. It is shown in [81] that θ^* is the solution to the projected Bellman equation. The following lemma shows that $(\theta^*; \theta^*)$ is also the GAS equilibrium point of the ODE of Double Q-learning. The reader is referred to the next subsection for the proof.

Lemma 10. $U^* = (\theta^*; \theta^*)$.

To apply Theorem 8, we need to work out $\nabla_\theta w(\theta^*), C_\theta(\theta^*), \nabla_U \tilde{w}(U^*)$, $C_U(U^*)$ which are the analogs of the quantities in Equation (5.20) for Q-learning and Double Q-learning, respectively. However, since the function H in Equation (5.22) could be non-differentiable around θ^*, we impose the following assumption from [90] that ensures the existence of $\nabla_\theta w(\theta^*)$ and $\nabla_U \tilde{w}(U^*)$.

Assumption 5. *The optimal policy $\pi^* := \pi_{\theta^*}$ is unique.*

Under this assumption, we summarize the exact forms of $\nabla_\theta w(\theta^*), C_\theta(\theta^*)$, $\nabla_U \tilde{w}(U^*), C_U(U^*)$ in the following result. The proof of this lemma is deferred to the next subsection.

Lemma 11. *Following the notation in the proof of Theorem 7, the following equations hold:*

$$\nabla_\theta w(\theta^*) = g\bar{A}, C_\theta(\theta^*) = g^2(B_1 + B_2), \tag{5.24a}$$

$$\nabla_U \tilde{w}(U^*) = g\bar{A}_D, C_U(U^*) = 2g^2 \begin{pmatrix} B_1 & B_2 \\ B_2 & B1 \end{pmatrix}, \tag{5.24b}$$

where $B_1 := \mathbb{E}\left[\sum_{n=1}^{\infty} W(Z_n)W(Z_1))^\top\right]$, $B_2 := \mathbb{E}\left[\sum_{n=2}^{\infty} W(Z_n)W(Z_1)^\top\right]$, *and* $W(Z_n) := (b(Z_n) + A_2(Z_n)\theta^* - A_1(Z_n)\theta^*)$.

Note that in Theorem 7, we assume $\theta^* = 0$. Therefore, $W(Z_n) = b(Z_n)$.

Define $g_0 := \inf\{g \geq 0 : g \max(\lambda_{\max}(\bar{A}), \lambda_{\max}(\bar{A}_D)) < -1\}$. Then whenever $g > g_0$, we have $\lambda_{\max}(\nabla_\theta w(\theta^*)) < -\frac{1}{2}, \lambda_{\max}(\nabla_U \tilde{w}(U^*)) < -\frac{1}{2}$. So far

we have checked all conditions in Theorem 8 for Q-learning and Double Q-learning. Therefore, the central limit theorem holds:

$$n^{\frac{1}{2}}(\theta_n - \theta^*) \underset{d}{\to} \mathcal{N}(0, P_Q) \tag{5.25a}$$

$$n^{\frac{1}{2}}(U_n - U^*) \underset{d}{\to} \mathcal{N}(0, P_D), \tag{5.25b}$$

where P_Q, P_D are given by

$$\left(\frac{I}{2} + g\bar{A}\right) P_Q + P_Q \left(\frac{I}{2} + g\bar{A}^\top\right) + g^2(B_1 + B_2) = 0 \tag{5.26a}$$

$$\left(\frac{I}{2} + g\bar{A}_D\right) P_D + P_D \left(\frac{I}{2} + g\bar{A}_D^\top\right) + 2g^2 \begin{pmatrix} B_1 & B_2 \\ B_2 & B_1 \end{pmatrix} = 0. \tag{5.26b}$$

We can see that Equations (5.26a) and (5.26b) are indeed identical to Equations (5.13) and (5.14) (for the asymptotic covariance matrices of Q-learning and Double Q-learning). However, since we only establish convergence in distribution of a sequence of random vectors, it does not immediately imply that the limit of variances of these random vectors converges to the variance of the corresponding normal distribution. To fix this gap, we first observe that the function $\mathbf{x}^\top \mathbf{x}$ is continuous in the vector $\mathbf{x}$. By the Continuous Mapping Theorem for random vectors and Equation (5.25), we have

$$n \left\|\theta_n - \theta^*\right\|_2^2 \underset{d}{\to} \left\|\mathbf{X}_Q\right\|_2^2 \tag{5.27a}$$

$$n \left\|(U_n - U^*)\right\|_2^2 \underset{d}{\to} \left\|\mathbf{X}_D\right\|_2^2, \tag{5.27b}$$

where $\mathbf{X}_Q$ follows the normal distribution $\mathcal{N}(0, P_Q)$, and $\mathbf{X}_D$ follows $\mathcal{N}(0, P_D)$. Here, the convergence in distribution is for random variables. Finally, to establish the convergence of the mean of these random variables, we need uniform integrability, which we assume as follows.

Assumption 6. *The three sequences of random variables*

$$\{n \left\|\theta_n - \theta^*\right\|_2^2, n \geq 1\}, \{n \left\|\theta_n^A - \theta^*\right\|_2^2, n \geq 1\}, \{n \left\|\theta_n^B - \theta^*\right\|_2^2, n \geq 1\}$$

are all uniformly integrable.

Assumption 6 directly implies the sequence $\{n \left\|U_n - U^*\right\|_2^2, n \geq 1\}$ is uni-

formly integrable. Combining Equation (5.27) with Assumption 6, we have

$$\lim_{n\to\infty} n\mathbb{E}\left[\|\theta_n - \theta^*\|_2^2\right] = \mathbb{E}\left[\|\mathbf{X}_Q\|_2^2\right] = \mathrm{Tr}(P_Q) \tag{5.28a}$$

$$\lim_{n\to\infty} n\mathbb{E}\left[\|(U_n - U^*)\|_2^2\right] = \mathbb{E}\left[\|\mathbf{X}_D\|_2^2\right] = \mathrm{Tr}(P_D). \tag{5.28b}$$

Under all the assumptions stated in this section, the linearizations in Section 5.2.3 are valid.

5.5.3 Proof of Lemmas

In this subsection, we present the proofs of Lemma 10 and Lemma 11.

Proof of Lemma 10: By Assumption 4, the ODE of Double Q-learning has a unique GAS equilibrium point. Denote this point as $(\theta_1; \theta_2)$. By the symmetry of the ODE in Equation (5.23), $(\theta_2; \theta_1)$ is also a GAS equilibrium point of the ODE. But such point is unique. We thus have $\theta_1 = \theta_2$. In this case, the ODE in Equation (5.23) degenerates to the ODE in Equation (5.22) of Q-learning. Therefore, we have $\theta_1 = \theta_2 = \theta^*$. □

Proof of Lemma 11: We prove the Q-learning result. A similar proof can be followed for the Double Q-learning result.

Recall the ODE of Q-learning defined in Equation (5.22). We know that θ^* is the unique GAS equilibrium point of this ODE. Recall that the right-hand side of Equation (5.22) is denoted by $w(\theta(t))$. Then at the point θ^*, the following equality holds:

$$w(\theta^*) = g\big(\mathbb{E}\left[\phi(X_n)R(X_n)\right] + \gamma\mathbb{E}\left[\phi(X_n)H(\theta^*, \theta^*, S_{n+1})\right] \\ - \mathbb{E}\left[\phi(X_n)\phi(X_n)^\top\right]\theta^*\big).$$

Note that the optimal policy π^* is unique by assumption. We can rewrite $H(\theta^*, \theta^*, S_{n+1})$ as $\phi(S_{n+1}, \pi^*(S_{n+1}))^\top\theta^*$. Therefore:

$$\begin{aligned} w(\theta^*) &= g\big(\mathbb{E}\left[\phi(X_n)R(X_n)\right] + \gamma\mathbb{E}\left[\phi(X_n)\phi(S_{n+1}, \pi^*(S_{n+1}))^\top\theta^*\right] \\ &\quad - \mathbb{E}\left[\phi(X_n)\phi(X_n)^\top\right]\theta^*\big) \\ &= g\mathbb{E}\left[\phi(X_n)R(X_n)\right] + g(\bar{A}_2 - \bar{A}_1)\theta^*, \end{aligned} \tag{5.29}$$

which is the same as the ODE of the linearization (see Equation (5.4)) at the point θ^*.

Furthermore, since the optimal policy is unique for θ^*, we can define:

$$\omega \coloneqq \min_{(s,a)\in\mathcal{X}:\ a\neq\pi^*(s)} (\phi(s,\pi^*(s))^\top\theta^* - \phi(s,a)^\top\theta^*) > 0$$

to be the minimum gap between value functions of optimal actions and non-optimal actions for all the states, estimated by θ^*. Let $\epsilon = \frac{\omega}{3\|\Phi\|_1}$. Consider any $\theta \in \mathbb{R}^d$ satisfying $\|\theta - \theta^*\|_\infty \leq \epsilon$. We claim that the greedy policy π_θ is equal to π^*. To show this, let us fix a state $s \in \mathcal{S}$. For any $a \in \mathcal{A}$ and $a \neq \pi^*(s)$, we have

$$\begin{aligned}\phi(s,\pi^*(s))^T\theta_a - \phi(s,a)^T\theta_a &\geq \phi(s,\pi^*(s))^T\theta^* - \phi(s,a)^T\theta^* - 2\left\|\Phi^T(\theta-\theta^*)\right\|_\infty \\ &\geq \omega - \frac{2\omega}{3} > 0.\end{aligned}$$

Therefore, $\pi_\theta = \pi^*$. Consequently, for any θ such that $\|\theta - \theta^*\|_\infty \leq \epsilon$, we have: $w(\theta) = g\mathbb{E}\left[\phi(X_n)R(X_n)\right] + g(\bar{A}_2 - \bar{A}_1)\theta$. Therefore, $\nabla_\theta w(\theta^*) = g\bar{A} = g(\bar{A}_2 - \bar{A}_1)$.

For $C_\theta(\theta^*)$, define:

$$W(Z_n) := \phi(X_n)R(X_n) + \gamma\phi(S_{n+1},\pi^*(S_{n+1}))\theta^* - \phi(X_n)\phi(X_n)^\top\theta^*.$$

Therefore, by definition

$$\begin{aligned}C_\theta(\theta^*) &= \sum_{n=-\infty}^{+\infty} \mathbb{E}\left[(gW(Z_n) - w(\theta^*))(gW(Z_1) - w(\theta^*))^\top\right] \\ &= g^2 \sum_{n=-\infty}^{+\infty} \mathbb{E}\left[(W(Z_n))(W(Z_1))^\top\right] \\ &= g^2\left(\sum_{n=1}^{+\infty} \mathbb{E}\left[(W(Z_n))(W(Z_1))^\top\right] + \sum_{n=2}^{+\infty} \mathbb{E}\left[(W(Z_n))(W(Z_1))^\top\right]\right).\end{aligned}$$

□

5.6 A Stronger Result for the Mean-Squared Error

In this section, we provide a stronger result for the asymptotic mean-squared error of Double Q-learning. Assume that the vector $b(x)$ defined in the proof of Theorem 7 is not the same for all $x \in \mathcal{X}$. Additionally, assume that $\theta^* = 0$. Following the notation in Theorem 7, we have the following result.

Theorem 9. *Let the step-sizes of Q-learning and Double Q-learning be $\alpha_n = g/n$ and $\delta_n = 2g/n$ respectively, where g is a positive constant. With the same constant g_0 in Theorem 7, for any $g > g_0$, we have*

$$\mathrm{AMSE}(\theta^A) \geq \mathrm{AMSE}(\theta) + c_0 g,$$

where c_0 is a positive constant independent from g.

Theorem 9 shows that in general, the asymptotic mean-squared error of Double Q-learning is worse than that of Q-learning, when using twice of the step-size. Moreover, the gap scales at least linearly with respect to the step-size.

To prove Theorem 9, we need two additional lemmas. The first lemma is on the relationship between the two matrices $\bar{A}_D$ and $\bar{A}$ defined in the proof of Theorem 7.

Lemma 12. *Following the notation in the proof of Theorem 7, consider the matrix $\bar{A}_D = \begin{pmatrix} -\bar{A}_1 & \bar{A}_2 \\ \bar{A}_2 & -\bar{A}_1 \end{pmatrix}$. The set of its eigenvalues is given by the union of the set of the eigenvalues of $\bar{A}_2 - \bar{A}_1$ and that of $-(\bar{A}_2 + \bar{A}_1)$.*

Proof of Lemma 12: Suppose λ is an eigenvalue of $\bar{A}_D$ with an eigenvector $v = (v_1^\top, v_2^\top)^\top \neq 0$ where $v_1, v_2 \in \mathbb{R}^d$. We claim that λ is either an eigenvalue of $-\bar{A}_1 + \bar{A}_2$ or an eigenvalue of $-(\bar{A}_1 + \bar{A}_2)$. To see this fact, we have

$$\bar{A}_D \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

If $v_1 + v_2 \neq \mathbf{0}$, then

$$(-\bar{A}_1 + \bar{A}_2)(v_1 + v_2) = \lambda(v_1 + v_2),$$

showing that λ is an eigenvalue of $-\bar{A}_1+\bar{A}_2$. Otherwise, suppose $v_1+v_2=\mathbf{0}$. Then $v_1=-v_2$, and

$$-(\bar{A}_1+\bar{A}_2)v_1=\lambda v_1.$$

We can also show that for every eigenvalue of $-\bar{A}_1+\bar{A}_2$ and $-(\bar{A}_1+\bar{A}_2)$, we can construct a corresponding eigenvector with respect to $\bar{A}_D$. Therefore, the set of the eigenvalues of $\bar{A}_D$ is exactly the union of the set of the eigenvalues of $-\bar{A}_1+\bar{A}_2$ and $-(\bar{A}_1+\bar{A}_2)$. □

The second lemma is on the trace of the solution of a Lyapunov equation.

Lemma 13. *Consider a Lyapunov equation*

$$AX+XA^\top+Q=0,$$

where $A,Q\in\mathbb{R}^{n\times n}$ are given, for some positive integer n. If A is Hurwitz, and $Q\succcurlyeq 0$, and $\mathrm{Tr}(Q)>0$, then $\mathrm{Tr}(X)>0$.

Note that the notation $Q\succcurlyeq 0$ means that Q is a positive semi-definite matrix.

Proof of Lemma 13: By [94, Theorem 5.6], if A is Hurwitz, then X has a unique solution that can be expressed as

$$X=\int_0^\infty e^{At}Qe^{A^\top t}\,\mathrm{d}t. \tag{5.30}$$

Since $Q\succcurlyeq 0$ by assumption, and $(e^{At})^\top=e^{A^\top t}$ for all t, we have $X\succcurlyeq 0$. We prove $\mathrm{Tr}(X)>0$ by contradiction. Suppose $\mathrm{Tr}(X)=0$. Therefore, as $X\succcurlyeq 0$, we have: $\mathbf{v}^\top X\mathbf{v}=0,\forall$ vectors $\mathbf{v}$ (since all eigenvalues of X are 0).

Denote the largest eigenvalue of Q as λ_m, which must be a positive real value because $Q\succcurlyeq 0$ and $\mathrm{Tr}(Q)>0$. Suppose $\mathbf{v}$ is the unit eigenvector corresponding to λ_m, i.e., $Q\mathbf{v}=\lambda_m\mathbf{v}$, and $\|\mathbf{v}\|_2=1$. We have

$$\mathbf{v}^\top X\mathbf{v}=\int_0^\infty \mathbf{v}^\top e^{At}Qe^{A^\top t}\mathbf{v}\,\mathrm{d}t. \tag{5.31}$$

Note that $\lim_{t\to 0}e^{At}=I$, and $\lim_{t\to 0}e^{A^\top t}=I$. Therefore, for $\epsilon=\min\left(\frac{\lambda_m}{\|Q\|_2},1\right)$, there exists a $\tilde{t}>0$, such that for any $0\le t\le\tilde{t}$, we have

$$\left\|e^{At}-I\right\|_2\le\epsilon,\qquad \left\|e^{A^\top t}-I\right\|_2\le\epsilon. \tag{5.32}$$

Equation (5.31) can be rewritten as

$$\begin{aligned}
\mathbf{v}^\top X \mathbf{v} &= \int_0^{\tilde{t}} \mathbf{v}^\top e^{At} Q e^{A^\top t} \mathbf{v} \, \mathrm{d}t + \int_{\tilde{t}}^{\infty} \mathbf{v}^\top e^{At} Q e^{A^\top t} \mathbf{v} \, \mathrm{d}t \\
&\overset{(a)}{\geq} \int_0^{\tilde{t}} \mathbf{v}^\top e^{At} Q e^{A^\top t} \mathbf{v} \, \mathrm{d}t \\
&= \int_0^{\tilde{t}} \mathbf{v}^\top (I + e^{At} - I) Q (I + e^{A^\top t} - I) \mathbf{v} \, \mathrm{d}t \\
&= \int_0^{\tilde{t}} \mathbf{v}^\top Q \mathbf{v} \, \mathrm{d}t + \int_0^{\tilde{t}} \mathbf{v}^\top (e^{At} - I) Q \mathbf{v} \, \mathrm{d}t + \int_0^{\tilde{t}} \mathbf{v}^\top Q (e^{A^\top t} - I) \mathbf{v} \, \mathrm{d}t \\
&\quad + \int_0^{\tilde{t}} \mathbf{v}^\top (e^{At} - I) Q (e^{A^\top t} - I) \mathbf{v} \, \mathrm{d}t.
\end{aligned} \tag{5.33}$$

Inequality (a) follows from the fact that $e^{At} Q e^{A^\top t} \succcurlyeq 0$, for any $t \geq 0$. To lower bound the right-hand side of the final equality in Equation (5.33), we first observe that $\int_0^{\tilde{t}} \mathbf{v}^\top Q \mathbf{v} \, \mathrm{d}t = \tilde{t} \, \|v\|_2^2 \, \lambda_m$, by the definition of $\mathbf{v}$. For the last three terms, using the definition of matrix norm and Equation (5.32), the following holds

$$\left| \int_0^{\tilde{t}} \mathbf{v}^\top (e^{At} - I) Q \mathbf{v} \, \mathrm{d}t \right| \leq \tilde{t} \, \|v\|_2^2 \, \|Q\|_2 \, \epsilon \tag{5.34}$$

$$\left| \int_0^{\tilde{t}} \mathbf{v}^\top Q (e^{A^\top t} - I) \mathbf{v} \, \mathrm{d}t \right| \leq \tilde{t} \, \|v\|_2^2 \, \|Q\|_2 \, \epsilon \tag{5.35}$$

$$\left| \int_0^{\tilde{t}} \mathbf{v}^\top (e^{At} - I) Q (e^{A^\top t} - I) \mathbf{v} \, \mathrm{d}t \right| \leq \tilde{t} \, \|v\|_2^2 \, \|Q\|_2 \, \epsilon^2. \tag{5.36}$$

Therefore, we have

$$\begin{aligned}
\mathbf{v}^\top X \mathbf{v} &\geq \tilde{t} \, \|v\|_2^2 \, \lambda_m - 2\tilde{t} \, \|v\|_2^2 \, \|Q\|_2 \, \epsilon - \tilde{t} \, \|v\|_2^2 \, \|Q\|_2 \, \epsilon^2 \\
&\geq \tilde{t} \, \|v\|_2^2 \left(\lambda_m - \|Q\|_2 \left(2\epsilon + \epsilon^2 \right) \right) \\
&\geq \frac{1}{2} \tilde{t} \, \|v\|_2^2 \, \lambda_m
\end{aligned} \tag{5.37}$$

by the definition of ϵ. We can see that $\mathbf{v}^\top X \mathbf{v} > 0$, which contradicts the assumption that $\mathbf{v}^\top X \mathbf{v} = 0$. Therefore, $\mathrm{Tr}(X) > 0$ by contradiction. □

We now present the proof of Theorem 9.

Proof of Theorem 9: This proof follows the notation in the proof of Theorem 7. In particular, we assume that the random vector $b(X_n)$ is centered at 0. Recall Equation (5.15). Subtracting the block on the upper left corner by that on the upper right corner, we have

$$\begin{aligned}(V-C)\left(\frac{1}{2}I - g(\bar{A}_1+\bar{A}_2)\right)^\top + \left(\frac{1}{2}I - g(\bar{A}_1+\bar{A}_2)\right)(V-C) \\ + 2g^2(B_1-B_2) = 0.\end{aligned} \tag{5.38}$$

By the definition of B_1 and B_2, we have $B_1 - B_2 = \mathbb{E}\left[b(X_1)b(X_1)^\top\right]$, which has a positive trace by the assumptions. As in the proof of Theorem 7, set the constant $g_0 := \inf\{g \geq 0 : g\max(\lambda_{\max}(\bar{A}), \lambda_{\max}(\bar{A}_D)) < -1\}$. Since the matrix $\bar{A}_D$ is defined as $\begin{pmatrix} -\bar{A}_1 & \bar{A}_2 \\ \bar{A}_2 & -\bar{A}_1 \end{pmatrix}$, we know by Lemma 12 that the set of eigenvalues of $-(\bar{A}_1+\bar{A}_2)$ is a subset of the set of eigenvalues of $\bar{A}_D$. Therefore, for $g > g_0$, we have $g\lambda_{\max}(-(\bar{A}_1+\bar{A}_2)) < -1$. It immediately implies that $\frac{1}{2}I - g(\bar{A}_1+\bar{A}_2)$ is Hurwitz. Utilizing Lemma 13, we have $\mathrm{Tr}(V-C) > 0$. Together with the result $V + C = 2\Sigma_\infty^Q$ in the proof of Theorem 7, we have

$$\mathrm{AMSE}(\theta^A) = \mathrm{Tr}(V) = \mathrm{Tr}(\Sigma_\infty^Q) + \frac{\mathrm{Tr}(V-C)}{2} > \mathrm{Tr}(\Sigma_\infty^Q) = \mathrm{AMSE}(\theta).$$

On the other hand, to show that $\mathrm{AMSE}(\theta^A) - \mathrm{AMSE}(\theta)$ indeed scales linearly with respect to g, we divide both sides of Equation (5.38) by g

$$(V-C)\left(\frac{1}{2g}I - (\bar{A}_1+\bar{A}_2)\right)^\top + \left(\frac{1}{2g}I - (\bar{A}_1+\bar{A}_2)\right)(V-C) + 2g(B_1-B_2) = 0.$$

Since $\frac{1}{2g}I - (\bar{A}_1+\bar{A}_2)$ is Hurwitz, the following equation has a unique positive definite solution X:

$$X\left(\frac{1}{2g}I - (\bar{A}_1+\bar{A}_2)\right)^\top + \left(\frac{1}{2g}I - (\bar{A}_1+\bar{A}_2)\right)X + (B_1-B_2) = 0$$

Therefore, $\mathrm{Tr}(V-C) = 2g\mathrm{Tr}(X)$. Further, let X' be the solution to the following Lyapunov equation

$$X'\left(-(\bar{A}_1+\bar{A}_2)\right)^\top + \left(-(\bar{A}_1+\bar{A}_2)\right)X' + (B_1-B_2) = 0.$$

Since $-(\bar{A}_1 + \bar{A}_2)$ is Hurwitz, and $B_1 - B_2$ has a positive trace, we have $\mathrm{Tr}(X') > 0$, which is independent of g. Using Equation (5.30) for X and X', it can be easily shown that $\mathrm{Tr}(X) \geq \mathrm{Tr}(X')$. This proves that $\mathrm{AMSE}(\theta^A) - \mathrm{AMSE}(\theta) \geq c_0 g$, for some positive constant c_0 independent from g. $\square$

CHAPTER 6

FUTURE DIRECTIONS

For each problem considered in this thesis, there are interesting future directions to pursue. We propose a few here:

1. For the structured multi-armed bandit problem, a more refined analysis can potentially close the gap between the upper bound and the lower bound.

2. For the bandits with two-level feedback problem, an investigation into the performance and analysis of Bayesian algorithms such as Thompson sampling can be of interest to the machine learning and networking community.

3. For the two-time scale reinforcement, an interesting question that arises is whether one can optimize the rate of convergence with respect to the time-scale ratio λ. Since the finite-time performance bound depends on a variety of problem-dependent parameters, it is difficult to optimize it over λ. An interesting direction of further research is to investigate if practical adaptive strategies for λ can be developed in order to improve the rate of convergence further.

4. For the Double Q-learning problem, there is some recent work on finite-time performance analysis [100]. Such results, along with potentially more refined analysis, can shed more light into how to adapt the learning rate and choose hyperparameters in order to speed up Double Q-learning's convergence rate.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *arXiv preprint arXiv:1312.5602*, 2013.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[5] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 465–472.

[6] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine, "Model-based value estimation for efficient model-free reinforcement learning," *arXiv preprint arXiv:1803.00101*, 2018.

[7] R. S. Sutton, "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming," in *Machine Learning Proceedings 1990*. Elsevier, 1990, pp. 216–224.

[8] T. Weber, S. Racanière, D. P. Reichert, L. Buesing, A. Guez, D. J. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li et al., "Imagination-augmented agents for deep reinforcement learning," *arXiv preprint arXiv:1707.06203*, 2017.

[9] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," in *Advances in Neural Information Processing Systems*, 2018, pp. 8224–8234.

[10] H. Gupta, A. Eryilmaz, and R. Srikant, "Link rate selection using constrained Thompson sampling," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 739–747.

[11] H. Gupta, R. Srikant, and L. Ying, "Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 4706–4715.

[12] W. Weng, H. Gupta, N. He, L. Ying, and R. Srikant, "The mean-squared error of double Q-learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[13] R. Combes, A. Proutiere, D. Yun, J. Ok, and Y. Yi, "Optimal rate sampling in 802.11 systems," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 2760–2767.

[14] R. Combes, J. Ok, A. Proutiere, D. Yun, and Y. Yi, "Optimal rate sampling in 802.11 systems: Theory, design, and implementation," *IEEE Transactions on Mobile Computing*, 2018.

[15] H. Gupta, A. Eryilmaz, and R. Srikant, "Low-complexity, low-regret link rate selection in rapidly time-varying wireless channels," in *INFOCOM, 2018 Proceedings IEEE*. IEEE, 2018.

[16] R. Combes and A. Proutiere, "Dynamic rate and channel selection in cognitive radio systems." *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 5, pp. 910–921, 2015.

[17] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, pp. 1–122, 2012.

[18] S. Paladino, F. Trovo, M. Restelli, and N. Gatti, "Unimodal Thompson sampling for graph-structured arms." in *AAAI*, 2017, pp. 2457–2463.

[19] J. C. Bicket, "Bit-rate selection in wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[20] A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[21] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: A practical approach," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2004, pp. 126–134.

[22] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*. ACM, 2008, pp. 118–131.

[23] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. ACM, 2001, pp. 236–251.

[24] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*. ACM, 2002, pp. 24–35.

[25] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[26] A. Garivier and O. Cappé, "The KL-UCB algorithm for bounded stochastic bandits and beyond," in *Proceedings of the 24th Annual Conference on Learning Theory*, 2011, pp. 359–376.

[27] S. Agrawal and N. Goyal, "Further optimal regret bounds for Thompson sampling," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 31. Scottsdale, Arizona, USA: PMLR, 29 Apr–01 May 2013, pp. 99–107.

[28] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[29] A. Gopalan, S. Mannor, and Y. Mansour, "Thompson sampling for complex online problems," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research. Bejing, China: PMLR, 22–24 Jun 2014, pp. 100–108.

[30] T. L. Graves and T. L. Lai, "Asymptotically efficient adaptive choice of control laws in controlled Markov chains," *SIAM Journal on Control and Optimization*, vol. 35, no. 3, pp. 715–743, 1997.

[31] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[32] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 107–110.

[33] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2693–2708, 2018.

[34] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[35] N. Kan, J. Zou, K. Tang, C. Li, N. Liu, and H. Xiong, "Deep reinforcement learning-based rate adaptation for adaptive 360-degree video streaming," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4030–4034.

[36] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "DRL360: 360-degree video streaming with deep reinforcement learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1252–1260.

[37] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: Optimizing 360 video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 394–407.

[38] T. Lattimore and C. Szepesvári, "Bandit algorithms," *Available online*, p. 28, 2018.

[39] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[40] S. Agrawal and N. Goyal, "Analysis of Thompson sampling for the multi-armed bandit problem," in *Conference on Learning Theory*, 2012, pp. 39–1.

[41] K. Cai, K. Chen, L. Huang, and J. C. Lui, "Multi-level feedback web links selection problem: Learning and optimization," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 763–768.

[42] K. Chen, K. Cai, L. Huang, and J. Lui, "Beyond the click-through rate: Web link selection with multi-level feedback," *arXiv preprint arXiv:1805.01702*, 2018.

[43] R. Combes, S. Magureanu, and A. Proutiere, "Minimal exploration in structured stochastic bandits," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 1763–1771. [Online]. Available: http://papers.nips.cc/paper/6773-minimal-exploration-in-structured-stochastic-bandits.pdf

[44] H. Gupta, A. Eryilmaz, and R. Srikant, "Low-complexity, low-regret link rate selection in rapidly-varying wireless channels," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 540–548.

[45] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1999.

[46] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Prentice Hall PTR, 1993.

[47] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[48] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena, 1996.

[49] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis lectures on Artificial Intelligence and Machine Learning*, vol. 4, no. 1, pp. 1–103, 2010.

[50] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, MA, 2011, vol. 2, no. 3.

[51] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth, *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*. Springer, 2012, vol. 434.

[52] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, 1997.

[53] S. Bhatnagar, D. Precup, D. Silver, R. S. Sutton, H. R. Maei, and C. Szepesvári, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *Advances in Neural Information Processing Systems*, 2009, pp. 1204–1212.

[54] R. S. Sutton, A. R. Mahmood, and M. White, "An emphatic approach to the problem of off-policy temporal-difference learning," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2603–2631, 2016.

[55] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. Springer Science & Business Media, 2012, vol. 22.

[56] H. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Science & Business Media, 2003, vol. 35.

[57] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Springer, 2009.

[58] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, "Finite sample analyses for TD(0) with function approximation," *arXiv preprint arXiv:1704.01161*, 2017, also appeared in AAAI 2018.

[59] G. Dalal, B. Szorenyi, G. Thoppe, and S. Mannor, "Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning," *arXiv preprint arXiv:1703.05376*, 2017, also appeared in COLT 2018.

[60] B. Liu, I. Gemp, M. Ghavamzadeh, J. Liu, S. Mahadevan, and M. Petrik, "Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity," *Journal of Artificial Intelligence Research*, vol. 63, pp. 461–494, 2018.

[61] C. Lakshminarayanan and C. Szepesvari, "Linear stochastic approximation: How far does constant step-size and iterate averaging go?" in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1347–1355.

[62] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," *arXiv preprint arXiv:1806.02450*, 2018.

[63] R. Srikant and L. Ying, "Finite-time error bounds for linear stochastic approximation and TD learning," *Conference on Learning Theory (COLT)*, 2019, arXiv preprint arXiv:1902.00923.

[64] R. S. Sutton, C. Szepesvári, and H. R. Maei, "A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation," *Advances in Neural Information Processing Systems*, vol. 21, no. 21, pp. 1609–1616, 2008.

[65] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 993–1000.

[66] V. R. Konda, J. N. Tsitsiklis et al., "Convergence rate of linear two-time-scale stochastic approximation," *The Annals of Applied Probability*, vol. 14, no. 2, pp. 796–819, 2004.

[67] P. Kokotovic, H. K. Khalil, and J. O'Reilly, *Singular Perturbation Methods in Control: Analysis and Design*. SIAM, 1999, vol. 25.

[68] H. K. Khalil, *Nonlinear Systems*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 3.

[69] H. Gupta, R. Srikant, and L. Ying, "Adaptive learning rate selection for temporal difference learning," *Real-world Sequential Decision Making Workshop, ICML*, 2019.

[70] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the Fourier basis," in *Twenty-fifth AAAI Conference on Artificial Intelligence*, 2011.

[71] H. V. Hasselt, "Double Q-learning," in *Advances in Neural Information Processing Systems*, 2010, pp. 2613–2621.

[72] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[73] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[74] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, Cambridge, UK, 1989.

[75] Z. Zhang, Z. Pan, and M. J. Kochenderfer, "Weighted Double Q-learning," in *Proc. Int. Jt. Conf. Artificial Intelligence (IJCAI)*, 2017, pp. 3455–3461.

[76] O. Anschel, N. Baram, and N. Shimkin, "Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning," in *Int. Conf. Machine Learning (ICML)*. PMLR, 2017, pp. 176–185.

[77] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. Springer Science & Business Media, 2012, vol. 22.

[78] J. N. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," in *Advances Neural Information Processing Systems (NeurIPS)*, 1997, pp. 1075–1081.

[79] V. S. Borkar and S. P. Meyn, "The ODE method for convergence of stochastic approximation and reinforcement learning," *SIAM J. Control Optim.*, vol. 38, no. 2, pp. 447–469, 2000.

[80] D. Lee and N. He, "Target-based temporal-difference learning," in *International Conference on Machine Learning*, 2019, pp. 3713–3722.

[81] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 664–671.

[82] D. Lee and N. He, "A unified switching system perspective and ODE analysis of Q-learning algorithms," *arXiv preprint arXiv:1912.02270*, 2019.

[83] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, "Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning," *Proceedings of Machine Learning Research vol*, vol. 75, pp. 1–35, 2018.

[84] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, "Finite sample analyses for TD(0) with function approximation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[85] C. Lakshminarayanan and C. Szepesvari, "Linear stochastic approximation: How far does constant step-size and iterate averaging go?" in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1347–1355.

[86] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," in *Conference On Learning Theory*, 2018, pp. 1691–1692.

[87] Z. Chen, S. Zhang, T. T. Doan, S. T. Maguluri, and J.-P. Clarke, "Performance of Q-learning with linear function approximation: Stability and finite-time analysis," *arXiv preprint arXiv:1905.11425*, 2019.

[88] Z. Chen, S. T. Maguluri, S. Shakkottai, and K. Shanmugam, "Finite-sample analysis of stochastic approximation using smooth convex envelopes," *arXiv preprint arXiv:2002.00874*, 2020.

[89] G. Qu and A. Wierman, "Finite-time analysis of asynchronous stochastic approximation and Q-learning," *arXiv preprint arXiv:2002.00260*, 2020.

[90] A. M. Devraj and S. P. Meyn, "Fastest convergence for Q-learning," *arXiv preprint arXiv:1707.03770*, 2017.

[91] A. M. Devraj and S. P. Meyn, "Q-learning with uniformly bounded variance: Large discounting is not a barrier to fast learning," *arXiv preprint arXiv:2002.10301*, 2020.

[92] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn, "Explicit mean-square error bounds for Monte-Carlo and linear stochastic approximation," *arXiv preprint arXiv:2002.02584*, 2020.

[93] B. Hu and U. Syed, "Characterizing the exact behaviors of temporal difference learning algorithms using Markov jump linear system theory," in *Advances in Neural Information Processing Systems*, 2019, pp. 8477–8488.

[94] C.-T. Chen, *Linear System Theory and Design*. Oxford University Press, Inc., 1998.

[95] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 30–37.

[96] A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, and J. P. How, "A tutorial on linear function approximators for dynamic programming and reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 6, no. 4, pp. 375–451, 2013.

[97] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 5, pp. 834–846, 1983.

[98] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.

[99] MC.AI, "OpenAI Gym's cart-pole balancing using Q-learning," https://mc.ai/openai-gyms-cart-pole-balancing-using-q-learning/, accessed August 3, 2020.

[100] H. Xiong, L. Zhao, Y. Liang, and W. Zhang, "Finite-time analysis for double Q-learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.